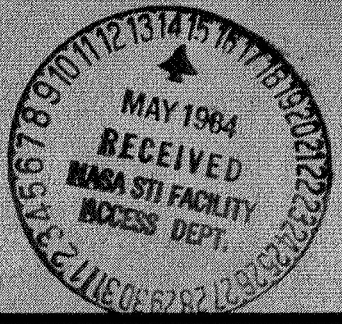
SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE REPORTING SOFTWARE USER'S GUIDE AND SYSTEM DESCRIPTION

VOLUME 2: PROGRAM DESCRIPTIONS

AUGUST 1983



National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

SOFTWARE ENGINEERING LABORATORY (SEL) DATA BASE REPORTING SOFTWARE USER'S GUIDE AND SYSTEM DESCRIPTION

VOLUME 2: PROGRAM DESCRIPTIONS

AUGUST 1983

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

FOREWORD

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration, Goddard Space Flight Center (NASA/GSFC) and created for the purpose of investigating the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1977 and has three primary organizational members:

NASA/GSFC (Systems Development and Analysis Branch)
The University of Maryland (Computer Sciences Department)
Computer Sciences Corporation (Flight Systems Operation)

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document. A version of this document was also issued as Computer Sciences Corporation document CSC/SD-82/6083-V1 and -V2.

The primary contributors to this document include

Pei-Shen Lo (Computer Sciences Corporation)
Suellen Eslinger (Computer Sciences Corporation)

Other contributors include

William Decker (Computer Sciences Corporation)

Single copies of this document can be obtained by writing to

Frank E. McGarry
Code 582.1
NASA/GSFC
Greenbelt, Maryland 20771

ABSTRACT

This two-volume document presents the Software Engineering Laboratory (SEL) data base reporting software user's guide and system description. The SEL data base reporting software programs provide formatted listings and summary reports of the SEL data base contents. This document is intended to serve as a reference or tool for the SEL data base administrator, librarians, and programmers and for managers and researchers involved in SEL data base activities. It describes the operating procedures and system information for 18 different reporting software programs.

Volume 1 contains an introduction summarizing the reporting software programs and detailed operating procedures for each program. Sample output reports from each program are also provided. Volume 2 contains descriptions of the structure and functions of each reporting software program. Baseline diagrams, module descriptions, and listings of program generation files are also included.

TABLE OF CONTENTS

VOLUME 1

TABLE OF CONTENTS (Cont'd)

Section 2 (Cont'd).

TABLE OF CONTENTS (Cont'd)

Section 2 (Cont'd)

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

Section 3 (Cont'd)

LIST OF ILLUSTRATIONS

Figure

Figure

Figure

LIST OF ILLUSTRATIONS (Cont'd)

Figure

LIST OF TABLES

Table

SECTION 3 - SYSTEM DESCRIPTION

This section contains the system descriptions for the SEL data base reporting programs. The function and structure of each program are presented. All accessed files are described, and, when applicable, baseline diagrams and descriptions of all routines in the program are provided. In addition, the task build procedure is described, including the command files, overlay structure, and required libraries.

3.1 DETAILED COMPONENT STATUS REPORT REPORTING PROGRAM (CS)

3.1.1 INTRODUCTION

The Detailed Component Status Report Reporting Program (CS) produces a report of the Component Status Report (CSR) file for a given project. The program provides a detailed breakdown of programmer hours as reported on the weekly CSR form for a given project (Section 2.1).

3.1.2 PROGRAM STRUCTURE

3.1.2.1 Files Accessed

The CS program accesses seven input files and two output files as described below.

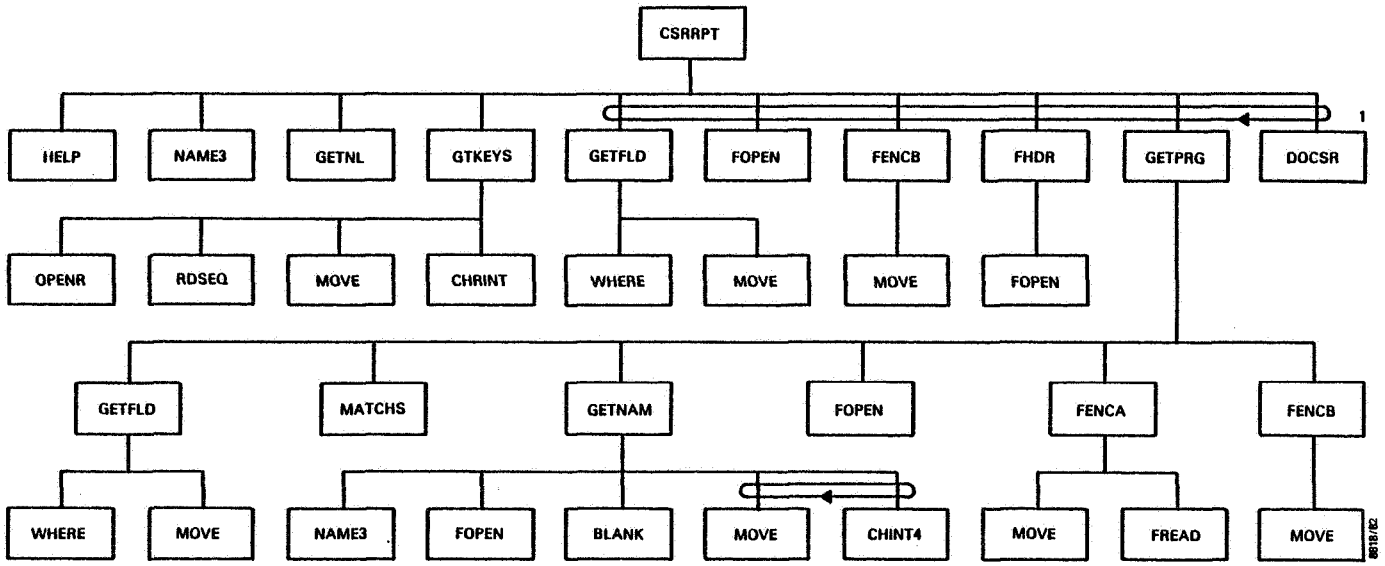
Input File Name	Description
[204,6]CSR.NL	CS parameters file
[204,6]CSR.KEY	CSR activity keywords file
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file
[204,1]HEADER.HDR	Phase Dates (HDR) file
[204,1]EST.HDR	Estimated Statistics (EST) file
[204,1]<PRJNAM>.CSR	CSR file for the given project
[204,1]<PRJNAM>.CIF	Component Information File (CIF) for the given project

Output File Name	Description
<PRJNAM>.CS	File containing the detailed CSR report
FOR010.DAT	File containing a list of all other activity names in file <PRJNAM>.CSR that did not match an activity subcategory name in file CSR.KEY

In these file names, <PRJNAM> is the name of the project selected by the user.

3.1.2.2 Baseline Diagram

Figure 3-1 is the baseline diagram for the CS program. The CSRRPT routine is the main driver. It displays the help



[1]SEE FIGURE 3-1 (2 OF 2).

Figure 3-1. Baseline Diagram for the Detailed Component Status Report Reporting Program (CS) (1 of 2)

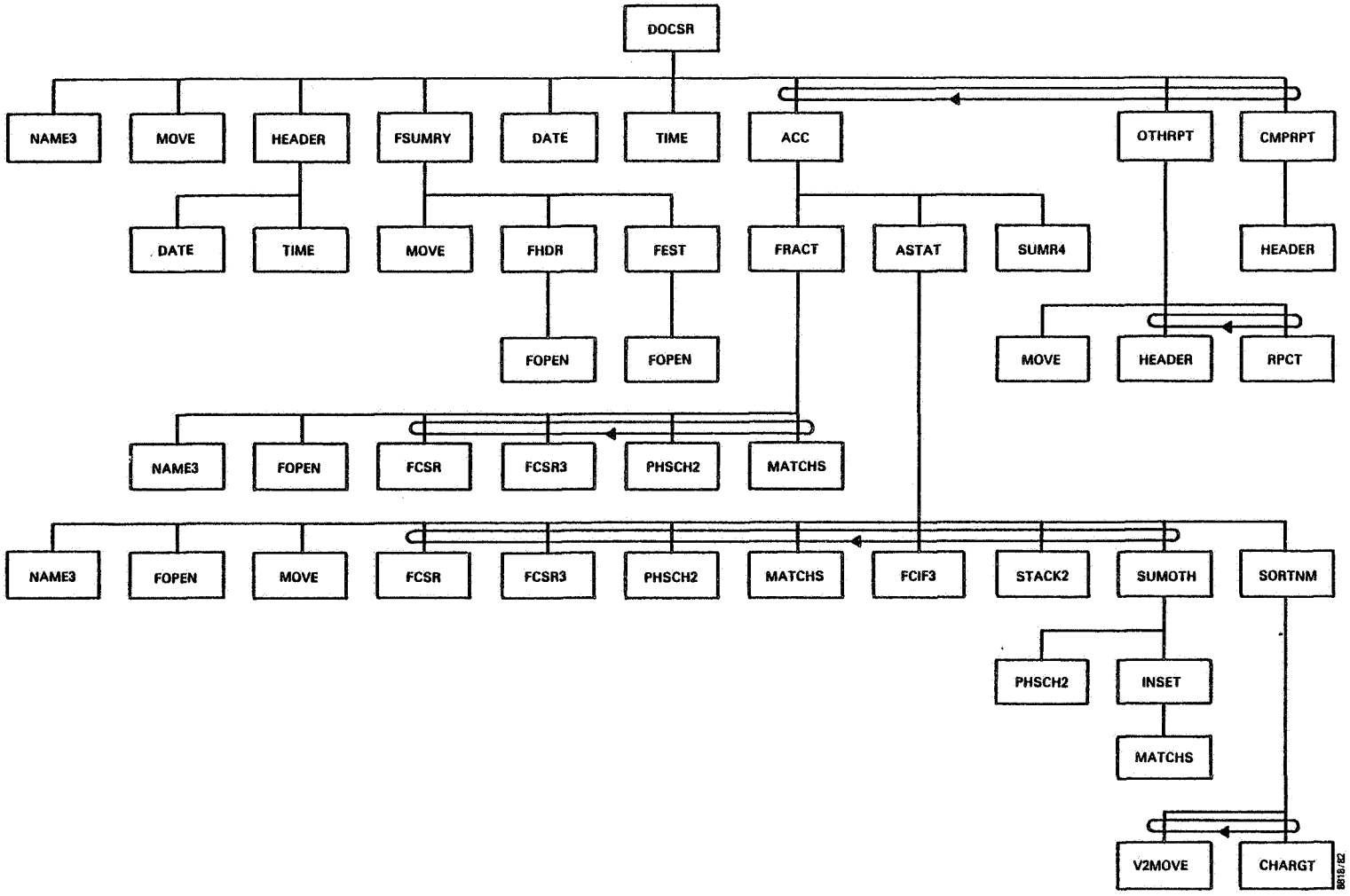


Figure 3-1. Baseline Diagram for the Detailed Component Status Report Reporting Program (CS) (2 of 2)

information, gets parameter values and other activity keyword values, obtains project and programmer names, and processes the selected CSR data. CSRRPT loops through this process until a^Z (control Z) is returned by the user in response to a prompt.

3.1.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the CS program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major CS routines are described in Section 3.1.3.8. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the CS program also uses the following system routines: DATE, ERRSET, and TIME.

3.1.3.1 Process CSR Data and Compute Statistics

These six major routines read the CSR file and accumulate statistics for the selected project and/or programmers.

ROUTINE: ACC

FUNCTION: Accumulates CSR statistics for a selected programmer

CALLING SEQUENCE:

```
CALL ACC (ACAT, ANAME, CSRFIL, FCOD, FDES, FTST, KEY,
          MAXCMP, MAXOTH, NCAT, NL, NNAME, NPROG, PROGCO,
          PRJNAM, RANGES, SOURCE, TOTFLG,

          A, ACOL, AOTH, AOTHTT, AROW, ASUBTT, ATOT, C,
          CCOL, CNAMES, CROW, CSIZE, CSORTX, CTOT, ERROR)
```

ROUTINE: ASTAT

FUNCTION: Accumulates component and other activity statistics by reading the CSR file

CALLING SEQUENCE:

```
CALL ASTAT (ANAME, CSRFIL, FCOD, FDES, FRREQ, FTST, KEY,
            MAXCMP, MAXOTH, NL, PRJNAM, PROGCO, RANGES,
            SOURCE, TOTFLG,
            A, AOTH, C, CNAMES, CSIZE, CSORTX, ERROR)
```

ROUTINE: CSRRPT

FUNCTION: Main routine of the CS program, produces the detailed CSR report

CALLING SEQUENCE: None

ROUTINE: DOCSR

FUNCTION: Processes CSR data by obtaining statistics and writes the CSR output report file

CALLING SEQUENCE:

```
CALL DOCSR (ACAT, ANAME, CATNAM, CSRFIL, KEY, MAXPRG,
            NCAT, NL, NNAME, NPROG, PRGCOD, PRGNAM,
            RANGES, RPTFIL, SOURCE, SUMARY)
```

ROUTINE: FRACT

FUNCTION: Reads the CSR file for the given project and computes the fraction of the design, code, and test phases for the given programmer

CALLING SEQUENCE:

```
CALL FRACT (CSRFIL, NL, PRJNAM, PROGCO, RANGES, TOTFLG,
            FCOD, FDES, FRREQ, FTST, TFRCOD, TFRDES,
            TFRTST, ERROR)
```

ROUTINE: SUMOTH

FUNCTION: Adds the time for a given other name to the appropriate statistics

CALLING SEQUENCE:

```
CALL SUMOTH (ANAME, FDATE, FCOD, FDES, FRREQ, FTST, KEY,
             MAXOTH, NL, OTHNAM, OTHOUR, RANGES, SOURCE,
             A, AOTH,
             FOUND)
```

3.1.3.2 Write the CSR Report File

These four routines write the CSR output report file.

ROUTINE: CMPRPT

FUNCTION: Prints the report section containing alphabetized component names and corresponding accumulated hours

CALLING SEQUENCE:

```
CALL CMPRPT (C, CNAMES, CSIZE, CSORTX, IPRG, MAXPRG, NL,
             PRGNAM, PRJNAM, RPTFIL, RPTITL, TOTFLG)
```

ROUTINE: FSUMRY

FUNCTION: Prints a six-line header summary with data from the Phase Dates and Estimated Statistics files

CALLING SEQUENCE:

```
CALL FSUMRY (IRPTF, PRJNAM)
```

ROUTINE: HEADER

FUNCTION: Prints a one-line title for each report page that includes the date and project name

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PRJNAM, RPTITL)
```

ROUTINE: OTHRPT

FUNCTION: Prints a report of other activity statistics

CALLING SEQUENCE:

```
CALL OTHRPT (A, ACAT, ACOL, ANAME, AOTH, AROW, ASUBTT,
             ATOT, CATNAM, IPRG, KEY, MAXOTH, MAXPRG,
             NCAT, NL, NNAME, PRGNAM, PRJNAM, RPTFIL,
             RPTITL, TOTFLG)
```

3.1.3.3 Obtain Data From Terminal or External File

These nine routines obtain information from a user's response to a terminal prompt or from an external file. This information includes input parameters, programmer names, other activity keyword names, and the project name.

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR,
             FIELD)
```

ROUTINE: GETNAM

FUNCTION: Gets all CSR programmer codes

CALLING SEQUENCE:

```
CALL GETNAM (CSRFIL, MAXPRG, PRJNAM,
             NPROG, PRGCOD, ERROR)
```

ROUTINE: GETNL

FUNCTION: Reads the sequential parameter file and fills the parameter array

CALLING SEQUENCE:

```
CALL GETNL (NLDSN, NLFIL, MAXNL,
            NL, ERROR)
```

ROUTINE: GETPRG

FUNCTION: Obtains programmer names from the user and converts them to programmer codes from the Encoding Dictionary

CALLING SEQUENCE:

```
CALL GETPRG (CSRFIL, EXTFIL, MAXPRG, NL, PRJNAM, TERMNL,
             NPROG, PRGCOD, PRGNAM, SUMARY, EOF, ERROR)
```

ROUTINE: GTKEYS

FUNCTION: Reads the sequential keywords file to obtain the necessary other activity names and keys for the detailed CSR report

CALLING SEQUENCE:

```
CALL GTKEYS (KEYFIL, MAXOTH, NL,
             ACAT, ANAME, CATNAM, KEY, NCAT, NNAME,
             SOURCE, ERROR)
```

ROUTINE: FENCA

FUNCTION: Finds the description field on the Encoding Dictionary corresponding to the given type and code

CALLING SEQUENCE:

```
CALL FENCA (IENCF, TYPE, CODE,
            NAME, REST, FOUND)
```

ROUTINE: FENCB

FUNCTION: Finds the description field on the Encoding Dictionary corresponding to the given type and name

CALLING SEQUENCE:

```
CALL FENCB (IENCF, TYPE, NAME,
            CODE, REST, FOUND)
```

ROUTINE: HELP

FUNCTION: Prints help information to the terminal

CALLING SEQUENCE:

```
CALL HELP
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

3.1.3.4 Sort and Search Routines

These four routines provide some sort and search functions.

ROUTINE: INSET

FUNCTION: Determines if the given eight-character name is in the given list of names

CALLING SEQUENCE:

```
CALL INSET (STRING, NAMES, MAXNAM,
            INDEX, FOUND)
```

ROUTINE: PHSCH2

FUNCTION: Determines to which phase the given date belongs

CALLING SEQUENCE:

```
CALL PHSCH2 (FDATE, RANGES,
             PHNUM, INPHAS)
```

ROUTINE: SORTNM

FUNCTION: Produces an array of pointers pointing to the given name array in alphabetical order

CALLING SEQUENCE:

```
CALL SORTNUM (NAMES, NDIM, NUSED, NAMLEN,
              SORTX)
```

ROUTINE: STACK2

FUNCTION: Determines whether the given name is in the given name array, adds it if it is not, and returns the location of the given name in the given name array

CALLING SEQUENCE:

```
CALL STACK2 (ARYMAX, NAME, NAMLEN, NL,
             ARY, ARYSIZ,
             LOC, MAXERR)
```

3.1.3.5 File Open and Read Routines

These nine routines either open an indexed file or read records from an indexed file.

ROUTINE: FCIF3

FUNCTION: Reads one record from the CIF using the tertiary key (component code) and converts all data to internal format

CALLING SEQUENCE:

```
CALL FCIF3 (ICIFF, CCODE,
            PROJNO, CNAME, ICODE, PANV, MODFUN, SYSFUN,
            ORIGIN, NEXEC, NLINES, NCOMNT, IETA1, IETA2,
            NETA1, NETA2, NIOVAR, MCCABE, NFUNCT, NIO,
            NASGN, NCALL, NFMT, STATUS, EOF, ERROR)
```

ROUTINE: FCSR

FUNCTION: Reads one record from the CSR file using a FORTRAN read

CALLING SEQUENCE:

```
CALL FCSR (ICSRF,
          FORMNO, SEQNO, PROJNO, PROGNO, FDATE, COMPCO,
          TIMES, OTHNAM, OTHOUR, ISTAT, PHASE, EOF,
          ERROR)
```

ROUTINE: FCSR3

FUNCTION: Reads one record from the CSR file using the tertiary key (programmer code)

CALLING SEQUENCE:

```
CALL FCSR3 (ICSRF, PROGCO,
           FORMNO, SEQNO, PROJNO, PROGNO, FDATE, COMPCO,
           TIMES, OTHNAM, OTHOUR, ISTAT, PHASE, EOF,
           ERROR)
```

ROUTINE: FEST

FUNCTION: Reads one record from the EST file using the secondary key (project name)

CALLING SEQUENCE:

```
CALL FEST (IESTF, NAME,
          PROJ, NCOMP, MODDEL, MODNEW, MODMOD, NRUNS,
          NCHANG, PAGDOC, LINDEL, LINNEW, LINMOD, TOTEXT,
          NEWEXT, MODEXT, PROGHR, MGMTHR, OTHRHR, HR95,
          HR75, OTHCMP, STATUS, ACTIVE, PRJCAT, FOUND,
          ERROR)
```

ROUTINE: FHDR

FUNCTION: Reads one record from the HDR file using the secondary key (project name)

CALLING SEQUENCE:

```
CALL FHDR (IHDRF, PRJNAM,
          PROJ, DEVCMP, TARG, ALIEN, RANGES, STATUS,
          ERROR)
```

ROUTINE: FOPEN

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL FOPEN (IUNIT, FILNAM,
           ERROR)
```

ROUTINE: FREAD

FUNCTION: Reads one indexed record

CALLING SEQUENCE:

```
CALL FREAD (IUNIT, KEYVAL, KEYLEN, LRECL,
           BUFFER, ERROR)
```

ROUTINE: OPENR

FUNCTION: Opens a sequential file for read only

CALLING SEQUENCE:

```
CALL OPENR (IUNIT, FILNAM, LEN,
           ERROR)
```

ROUTINE: RDSEQ

FUNCTIONS: Reads one record in a sequential file

CALLING SEQUENCE:

```
CALL RDSEQ (IUNIT, NCHAR,
           CHARS, EOF)
```

3.1.3.6 Routines for String Movement or Comparison

These eight routines deal with string movement or comparison.

ROUTINE: BLANK

FUNCTION: Initializes an array to blanks

CALLING SEQUENCE:

```
CALL BLANK (ARRAY, NUM)
```

ROUTINE: CHARGT (LOGICAL FUNCTION)

FUNCTION: Determines if the first string follows the second in alphabetical order

CALLING SEQUENCE:

```
CHARGT (STRNG1, STRNG2, LEN)
```

ROUTINE: CHINT4

FUNCTION: Converts the given character string to an I*4 integer

CALLING SEQUENCE:

```
CALL CHINT4 (CHARS, NCHAR,
             I4NUM, ERROR)
```

ROUTINE: CHRINT

FUNCTION: Converts the given character string to an I*2 integer

CALLING SEQUENCE:

```
CALL CHRINT (CHARS, NCHAR,
             I2NUM, ERROR)
```

ROUTINE: MATCHS (LOGICAL FUNCTION)

FUNCTION: Determines if the two input strings are the same

CALLING SEQUENCE:

```
MATCHS (ARRAY1, ARRAY2, NBYTES)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: V2MOVE

FUNCTION: Copies bytes from one row of a virtual array to a nonvirtual character string

CALLING SEQUENCE:

```
CALL V2MOVE (ARY2D, STRING, NROW, DIM1, DIM2)
```

ROUTINE: WHERE

FUNCTION: Finds the location of the given character in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
            LOC, FOUND)
```

3.1.3.7 Mathematical Functions

These two routines perform mathematical functions.

ROUTINE: RPCT (REAL FUNCTION)

FUNCTION: Computes a percentage

CALLING SEQUENCE:

RPCT (I, J)

ROUTINE: SUMR4 (REAL FUNCTION)

FUNCTION: Computes the sum of all numbers in a given array

CALLING SEQUENCE:

SUMR4 (ARRAY, N)

3.1.3.8 Variable Description

The variables in the calling sequences of major CS routines are described below.

Name	Type	Description
A(6,MAXOTH)	R*4	Array containing hours spent on other activities during each phase
ACAT(MAXOTH)	I*2	Activity name category array to indicate which category the given name belongs to
ACOL(6)	R*4	Array containing column total of each phase
ANAME(12,MAXOTH)	L*1	Array containing other activity names
AOTH(6)	R*4	Array containing hours spent on unknown activities that were not on the list of ANAME for each phase
AOTHTT	I*2	Not used
AROW(MAXOTH)	R*4	Array containing total hours spent on each activity
ARY(NAMLEN, ARYMAX)	L*1	Name array to be searched
ARYMAX	I*2	Maximum number of names in ARY
ARYSIZ	I*2	Actual number of names in ARY
ASUBTT(6,20)	R*4	Array containing total hours spent on each category for each phase
ATOT	I*2	Not used
C(9,MAXCMP)	R*4	Array containing hours spent on a component during different phases

Name	Type	Description
CATNAM(20,20)	L*1	Array containing category names for other activities
CCOL(3)	I*2	Not used
CNAMES(8,MAXCMP)	L*1	Array containing component names
CROW(MAXCMP)	I*2	Not used
CSIZE	I*2	Total number of components
CSORTX(MAXCMP)	I*2	Array containing index for sorted component names
CSRFIL	I*2	FORTRAN unit number for CSR file
CTOT	I*2	Not used
EOF	L*1	End-of-file flag
EOFTTY	L*1	Flag for end of file on terminal
ERROR	L*1	Error flag
EXTFIL	I*2	FORTRAN unit number for reading user input from terminal
FCOD	R*4	Fraction of time a given programmer spent on coding
FDATE(3)	I*2	Form date (YY,MM,DD)
FDES	R*4	Fraction of time a given programmer spent on design
FIELD(FLDLEN)	L*1	Field to be obtained
FLDLEN	I*2	Length of field
FOUND	L*1	Flag indicating a given name is found
FRREQ(7)	R*4	Fraction of time a given programmer spent on other activities during each phase
FTST	R*4	Fraction of time a given programmer spent on testing
INDEX	I*2	Location of a given name within an array of names
INPHAS	L*1	Flag indicating if a given form date is in any phase
IPRG	I*2	Current programmer number
IRPTF	I*2	FORTRAN unit number for CS output report file

Name	Type	Description
KEY(MAXOTH)	I*2	Array containing keywords for other activity names
KEYFIL	I*2	FORTRAN unit number for data set CSR.KEY
LOC	I*2	Location of a given name in the given name array
MAXCMP	I*2	Maximum number of components
MAXERR	L*1	Flag indicating whether the maximum number of components is exceeded
MAXNAM	I*2	Maximum number of other activity names
MAXNL	I*2	Maximum number of input parameters
MAXOTH	I*2	Maximum number of other activity names
MAXPRG	I*2	Maximum number of programmers
NAME(NAMLEN)	L*1	Given name to be searched for
NAMES(NAMLEN, NDIM)	L*1	Name array to be sorted or to be searched
NAMLEN	I*2	Length of the name
NCAT	I*2	Number of name categories
NDIM	I*2	Maximum number of names
NL(MAXNL)	I*2	Array containing input parameter values
NLDSN(27)	L*1	Input parameter file name
NLFIL	I*2	FORTRAN unit number for the input parameter file (CSR.NL)
NNAME	I*2	Total number of other activity names
NPROG	I*2	Total number of programmers
NUSED	I*2	The actual number of names (fill size of NAMES)
OTHNAM(8)	L*1	Other activity name
OTHOUR	R*4	Other activity work hours
PHNUM	I*2	Number of phase containing date
PRGCOD(MAXPRG)	I*4	Array containing programmer's code
PRGNAM(8,MAXPRG)	L*1	Array containing programmer's name

Name	Type	Description
PRJNAM(8)	L*1	Project name
PROGCO	I*4	Given programmer's code
RANGES(3,2,7)	I*2	Start and stop phase dates (YY,MM,DD)
RPTFIL	I*2	FORTRAN unit number for CS output report file
RPTITL(40)	L*1	Report title
SORTX(NDIM)	I*2	Array containing index for sorted names
SOURCE(MAXOTH)	L*1	Array containing keywords for other activity names
STRING(8)	L*1	Name string
SUMARY	L*1	Flag indicating whether a summary report is needed
TERMNL	L*1	Flag indicating whether terminal or external file is to be read
TEXT(FLDLEN)	L*1	Prompt text string
TFRCOD	R*4	Fraction of total time spent on coding
TFRDES	R*4	Fraction of total time spent on design
TFRTST	R*4	Fraction of total time spent on testing
TOTFLG	L*1	Flag indicating whether processing is for all programmers

3.1.4 TASK BUILD PROCEDURE

3.1.4.1 Command Procedures

The CS program can be generated from the source code by executing the command procedure CSGEN.CMD under UIC [204,6]. This command procedure references three command files--CSFPP.CMD, CSFOR.CMD, and CS.TKB--all under UIC [204,6]. Figure 3-2 is a listing of CSGEN.CMD, the command procedure to precompile, compile, and task build the CS program. The CS program is generated by entering the following command:

```
@[204,6]CSGEN
```

```
;                                                                          1
;    @CSGEN.CMD                                                            2
;                                                                          3
;    THIS COMMAND PROCEDURE GENERATES THE CS TASK FROM STRUCTURED           4
;    FORTRAN SOURCE.                                                       5
;                                                                          6
;    PRECOMPILE STRUCTURED FORTRAN SOURCE                                  7
;                                                                          8
@[204,6]CSFPP                                                              9
;                                                                         10
;     @CSFPP.CMD                                                          11
;                                                                         12
;     THIS COMMAND PROCEDURE PRECOMPILES ALL ROUTINES WHICH CS PROGRAM     13
;     USES. ALL ROUTINES ARE WRITTEN IN STRUCTURED FORTRAN.               14
;                                                                         15
;     ALL ROUTINES WITH PREFIX CS                                         16
;                                                                         17
;FPP SY:[204,6]CSACC                                                      18
;FPP SY:[204,6]CSASTAT                                                    19
;FPP SY:[204,6]CSCMPRPT                                                   20
;FPP SY:[204,6]CSCSRRPT                                                   21
;FPP SY:[204,6]CSDOCSR                                                    22
;FPP SY:[204,6]CSFRACT                                                    23
;FPP SY:[204,6]CSGETNAM                                                   24
;FPP SY:[204,6]CSGETNL                                                    25
;FPP SY:[204,6]CSGETPRG                                                   26
;FPP SY:[204,6]CSGTKEYS                                                   27
;FPP SY:[204,6]CSHELP                                                     28
;FPP SY:[204,6]CSINSET                                                    29
;FPP SY:[204,6]CSOTHRPT                                                   30
;FPP SY:[204,6]CSSORTNM                                                   31
;FPP SY:[204,6]CSSTACK2                                                   32
;FPP SY:[204,6]CSSUMOTH                                                   33
;                                                                         34
;     ROUTINE WITH PREFIX NF                                              35
;                                                                         36
;FPP SY:[204,6]NFSUM                                                      37
;                                                                         38
;     ROUTINES WITH PREFIX UT                                             39
;                                                                         40
;FPP SY:[204,7]UTBLANK                                                    41
;FPP SY:[204,7]UTCHARGT                                                   42
;FPP SY:[204,7]UTCHINT4                                                   43
;FPP SY:[204,7]UTCHRINT                                                   44
;FPP SY:[204,7]UTFCIF3                                                    45
;FPP SY:[204,7]UTFCSR                                                     46
;FPP SY:[204,7]UTFCSR3                                                    47
;FPP SY:[204,7]UTFENCA                                                    48
;FPP SY:[204,7]UTFENCB                                                    49
;FPP SY:[204,7]UTFEST                                                     50
;FPP SY:[204,7]UTFHDR                                                     51
;FPP SY:[204,7]UTFOPEN                                                    52
;FPP SY:[204,7]UTFREAD                                                    53
;FPP SY:[204,7]UTFSUMRY                                                   54
;FPP SY:[204,7]UTGETFLD                                                   55
```

Figure 3-2. CS Task Generation Command Procedure (CSGEN.CMD) (1 of 3)

```
;FPP SY:[204,7]UTHEADER                                                    56
;FPP SY:[204,7]UTMATCHS                                                    57
;FPP SY:[204,7]UTMOVE                                                      58
;FPP SY:[204,7]UTNAME3                                                     59
;FPP SY:[204,7]UTPHSCH2                                                    60
;FPP SY:[204,7]UTRDSEQ                                                     61
;FPP SY:[204,7]UTRPCT                                                      62
;FPP SY:[204,7]UTSUMR4                                                     63
;FPP SY:[204,7]UTV2MOVE                                                    64
;FPP SY:[204,7]UTWHERE                                                     65
;                                                                         66
;    COMPILE FORTRAN SOURCE                                               67
;                                                                         68
@[204,6]CSFOR                                                             69
;                                                                         70
;   @CSFOR.CMD                                                            71
;                                                                         72
;   THIS COMMAND PROCEDURE COMPILES ALL FORTRAN ROUTINES WHICH CS         73
;   PROGRAM USES.                                                         74
;                                                                         75
;   ROUTINES WITH PREFIX CS                                               76
;                                                                         77
;FOR/F4P/OBJECT:[204,6]CSACC     [204,6]CSACC                              78
;FOR/F4P/OBJECT:[204,6]CSASTAT   [204,6]CSASTAT                            79
;FOR/F4P/OBJECT:[204,6]CSCMPRPT  [204,6]CSCMPRPT                           80
;FOR/F4P/OBJECT:[204,6]CSCSRRPT  [204,6]CSCSRRPT                           81
;FOR/F4P/OBJECT:[204,6]CSDOCSR   [204,6]CSDOCSR                            82
;FOR/F4P/OBJECT:[204,6]CSFRACT   [204,6]CSFRACT                            83
;FOR/F4P/OBJECT:[204,6]CSGETNAM  [204,6]CSGETNAM                           84
;FOR/F4P/OBJECT:[204,6]CSGETNL   [204,6]CSGETNL                            85
;FOR/F4P/OBJECT:[204,6]CSGETPRG  [204,6]CSGETPRG                           86
;FOR/F4P/OBJECT:[204,6]CSGTKEYS  [204,6]CSGTKEYS                           87
;FOR/F4P/OBJECT:[204,6]CSHELP    [204,6]CSHELP                             88
;FOR/F4P/OBJECT:[204,6]CSINSET   [204,6]CSINSET                            89
;FOR/F4P/OBJECT:[204,6]CSOTHRPT  [204,6]CSOTHRPT                           90
;FOR/F4P/OBJECT:[204,6]CSSORTNM  [204,6]CSSORTNM                           91
;FOR/F4P/OBJECT:[204,6]CSSTACK2  [204,6]CSSTACK2                           92
;FOR/F4P/OBJECT:[204,6]CSSUMOTH  [204,6]CSSUMOTH                           93
;                                                                         94
;   ROUTINE WITH PREFIX NF                                                95
;                                                                         96
;FOR/F4P/OBJECT:[204,6]NFSUM     [204,6]NFSUM                              97
;                                                                         98
;   ROUTINES WITH PREFIX UT                                               99
;                                                                        100
;FOR/F4P/OBJECT:[204,7]UTBLANK   [204,7]UTBLANK                           101
;FOR/F4P/OBJECT:[204,7]UTCHARGT  [204,7]UTCHARGT                          102
;FOR/F4P/OBJECT:[204,7]UTCHINT4  [204,7]UTCHINT4                          103
;FOR/F4P/OBJECT:[204,7]UTCHRINT  [204,7]UTCHRINT                          104
;FOR/F4P/OBJECT:[204,7]UTFCIF3   [204,7]UTFCIF3                           105
;FOR/F4P/OBJECT:[204,7]UTFCSR    [204,7]UTFCSR                            106
;FOR/F4P/OBJECT:[204,7]UTFCSR3   [204,7]UTFCSR3                           107
;FOR/F4P/OBJECT:[204,7]UTFENCA   [204,7]UTFENCA                           108
;FOR/F4P/OBJECT:[204,7]UTFENCB   [204,7]UTFENCB                           109
;FOR/F4P/OBJECT:[204,7]UTFEST    [204,7]UTFEST                            110
```

Figure 3-2. CS Task Generation Command Procedure (CSGEN.CMD) (2 of 3)

```
;FOR/F4P/OBJECT:[204,7]UTFHDR   [204,7]UTFHDR                              111
;FOR/F4P/OBJECT:[204,7]UTFOPEN  [204,7]UTFOPEN                            112
;FOR/F4P/OBJECT:[204,7]UTFREAD  [204,7]UTFREAD                            113
;FOR/F4P/OBJECT:[204,7]UTFSUMRY [204,7]UTFSUMRY                           114
;FOR/F4P/OBJECT:[204,7]UTGETFLD [204,7]UTGETFLD                           115
;FOR/F4P/OBJECT:[204,7]UTHEADER [204,7]UTHEADER                           116
;FOR/F4P/OBJECT:[204,7]UTMATCHS [204,7]UTMATCHS                           117
;FOR/F4P/OBJECT:[204,7]UTMOVE   [204,7]UTMOVE                             118
;FOR/F4P/OBJECT:[204,7]UTNAME3  [204,7]UTNAME3                            119
;FOR/F4P/OBJECT:[204,7]UTPHSCH2 [204,7]UTPHSCH2                           120
;FOR/F4P/OBJECT:[204,7]UTRDSEQ  [204,7]UTRDSEQ                            121
;FOR/F4P/OBJECT:[204,7]UTRPCT   [204,7]UTRPCT                             122
;FOR/F4P/OBJECT:[204,7]UTSUMR4  [204,7]UTSUMR4                            123
;FOR/F4P/OBJECT:[204,7]UTV2MOVE [204,7]UTV2MOVE                           124
;FOR/F4P/OBJECT:[204,7]UTWHERE  [204,7]UTWHERE                            125
;                                                                        126
;    GENERATE THE CS TASK IMAGE                                          127
;                                                                        128
TKB @[204,6]CS.TKB                                                       129
;                                                                        130
;   @CS.TKB                                                              131
;                                                                        132
;   THIS COMMAND PROCEDURE BUILDS A TASK IMAGE FOR THE DETAILED          133
;   COMPONENT STATUS REPORT PROGRAM (CS).                                134
;                                                                        135
:[204,5]CS=[204,6[ ]CS/MP                                                136
:UNITS=11                                                                137
;ACTFIL=8                                                                138
;//                                                                      139
;                                                                        140
```

Figure 3-2. CS Task Generation Command Procedure (CSGEN.CMD) (3 of 3)

3.1.4.2 Overlay Structure

The CS program is overlaid to reduce the memory space requirement. Figure 3-3 is a listing of the Overlay Descriptor Language file, [204,6]CS.ODL, needed to build the CS program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @CS.ODL                                                                2
;                                                                          3
;   THE OVERLAY STRUCTURE FOR THE DETAILED COMPONENT STATUS REPORT         4
;   PROGRAM (CS)                                                           5
;                                                                          6
          .ROOT $TREE1,OTSALL,RMSALL                                       7
$TREE1:   .FCTR $ROOT-RMSROT-OTSROT-*($HLP,$NL,$KEY,$PRG,$DO)              8
$ROOT:    .FCTR [204, 6]CSCSRRPT-[204, 7]UTMOVE  -[204, 7]UTNAME3 -$ROT4   9
$ROT4:    .FCTR [204, 7]UTMATCHS-$ROT6                                     10
$ROT6:    .FCTR [204, 7]UTOPENR -[204, 7]UTRDSEQ -$ROT8                    11
$ROT8:    .FCTR [204, 7]UTFENCA -[204, 7]UTFENCB -[204, 7]UTBLANK -$ROT12  12
$ROT12:   .FCTR [204, 7]UTFREAD -[204, 7]UTGETFLD-[204, 7]UTWHERE -$ROT14  13
$ROT14:   .FCTR [204, 7]UTHEADER-[204, 7]UTFHDR  -[204, 7]UTFOPEN          14
;                                                                          15
$HLP:     .FCTR [204, 6]CSHELP                                             16
;                                                                          17
$NL:      .FCTR [204, 6]CSGETNL                                            18
;                                                                          19
$KEY:     .FCTR [204, 6]CSGTKEYS-[204, 7]UTCHRINT                          20
;                                                                          21
$PRG:     .FCTR [204, 6]CSGETPRG-[204, 6]CSGETNAM-$PRG2                    22
$PRG2:    .FCTR [204, 7]UTCHINT4                                           23
;                                                                          24
;$DO:     .FCTR [204, 6]CSDOCSR -$RCSR-$RCSR3-$DO2                         25
$DO:      .FCTR [204, 6]CSDOCSR -$DO2                                      26
$DO2:     .FCTR ($HED,$FR,$ACC,$ORPT,$CRPT)                                27
;                                                                          28
$HED:     .FCTR [204, 7]UTFSUMRY-[204, 7]UTFEST                            29
;                                                                          30
$ACC:     .FCTR [204, 6]CSACC   -($FR,$AS,$SUM)                            31
;                                                                          32
$FR:      .FCTR [204, 6]CSFRACT -[204, 7]UTPHSCH2-$FR2                     33
$FR2:     .FCTR [204, 7]UTSUMR4 -[204, 7]UTCHINT4-$FR3                     34
$FR3:     .FCTR ([204,7]UTFCSR, [204, 7]UTFCSR3)                           35
;                                                                          36
$AS:      .FCTR [204, 6]CSASTAT -[204, 6]CSSUMOTH-$PHS-$AS2                37
$AS2:     .FCTR ($INS,$RCSR,$RCSR3,$CIF,$STK,$SORT)                        38
$PHS:     .FCTR [204, 7]UTPHSCH2                                           39
$INS:     .FCTR [204, 6]CSINSET                                            40
$RCSR:    .FCTR [204, 7]UTFCSR                                             41
$RCSR3:   .FCTR [204, 7]UTFCSR3                                            42
$CIF:     .FCTR [204, 7]UTFCIF3                                            43
$STK:     .FCTR [204, 6]CSSTACK2                                           44
$SORT:    .FCTR [204, 6]CSSORTNM-[204, 7]UTCHARGT-[204, 7]UTV2MOVE         45
;                                                                          46
;                                                                          47
$SUM:     .FCTR [204, 6]NFSUM                                              48
;                                                                          49
$ORPT:    .FCTR [204, 6]CSOTHRPT-[204, 7]UTRPCT                            50
;                                                                          51
$CRPT:    .FCTR [204, 6]CSCMPRPT                                           52
;                                                                          53
;                                                                          54
@LB:[1,1]RMS11M.ODL                                                        55
@LB:[1,1]RMS12X.ODL                                                        56
    .END                                                                   57
```

Figure 3-3. CS Program Overlay Descriptor Language File (CS.ODL)

3.2 PROFILE REPORT PROGRAM (PF)

3.2.1 INTRODUCTION

The Profile Report Program (PF) (or Generalized Response Accumulator Program) produces a cross-tabulation (or profile) report of the entries in various fields of a selected SEL data base file. The program supports the Component Information File (CIF), the Change Report Form (CRF) file, the Component Summary Form (CSF) file, and the Run Analysis Form (RAF) file.

3.2.2 PROGRAM STRUCTURE

3.2.2.1 Files Accessed

The PF program accesses two input files and one or more output files, depending on the file type selected. These files are described below.

Input File Name	Description
[204,6]PFNL.XXX	A sequential file containing the PF description file (Section 2.2.2), where XXX = file type (CIF, CRF, CSF, or RAF)
[204,1]<PRJNAM>.XXX	SEL data base file for the given project, where XXX = file type (CIF, CRF, CSF, or RAF)

Output File Name	Description
<PRJNAM>.YNN	The profile report file for the given project, where Y = report type (I, H, M, or A) and NN = breakdown variable number (Section 2.2.3)
<PRJNAM>.NNY	The plot file for the given project, where Y = report type (I, H, M, or A) and NN = breakdown variable number (Section 2.2.3); produced only for certain file types and breakdown variables (Section 2.2.3)

In these file names, <PRJNAM> is the name of the project selected by the user.

3.2.2.2 Baseline Diagram

Figure 3-4 is the baseline diagram for the PF program. The PROFIL routine is the main driver. It obtains the user's choices for project name, report type, and breakdown category; reads the selected file; accumulates responses; and writes the report. The driver loops through this process until a^Z (control Z) is returned by the user in response to a prompt.

3.2.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the PF program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major PF routines are described in Section 3.2.3.6. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the PF program also uses the following system routines: DATE, ERRSET, ERRSNS, and TIME.

3.2.3.1 Process Data and Accumulate Responses

These six major routines read a given data base file and accumulate responses for the specified profile report.

ROUTINE: GETDAT

FUNCTION: Reads the desired file and accumulates all statistics

CALLING SEQUENCE:

```
CALL GETDAT (BRKVAR, CATSIZ, DBFILE, FILTYP, IDBF, NCAT,
             RANGES, RINDEX, RNGCHK, VARNUM,
             K, KTOT, ERROR)
```

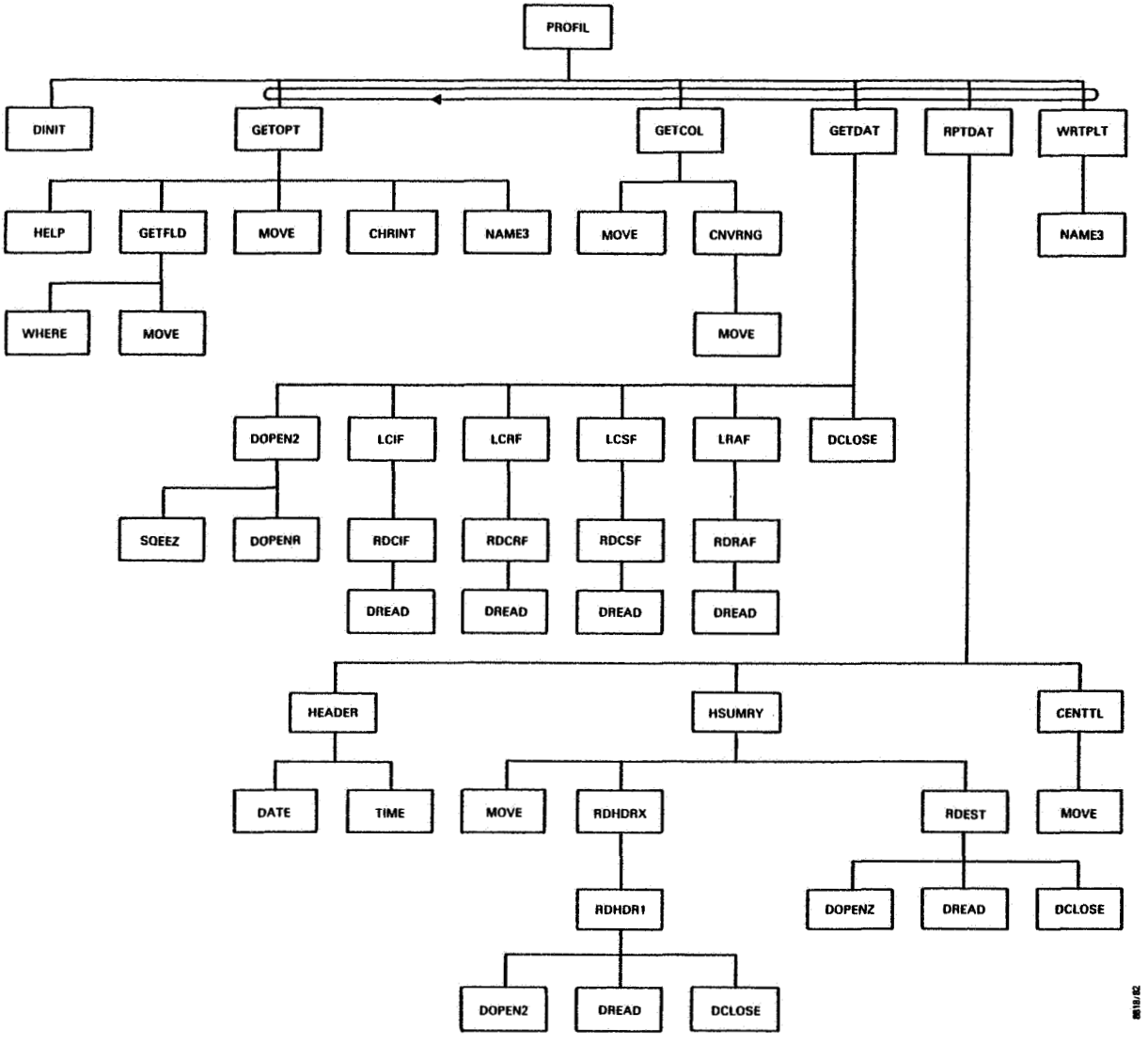


Figure 3-4. Baseline Diagram for the Profile Report Program (PF)

ROUTINE: LCIF

FUNCTION: Reads one record from the CIF and converts all significant data to an interger value from 1 to N

CALLING SEQUENCE:

```
CALL LCIF (ICIFF, RANGES, RINDEX,
          L, NULL, EOF, ERROR)
```

ROUTINE: LCRF

FUNCTION: Reads one record from the CRF file and converts all significant data to an integer value from 1 to N

CALLING SEQUENCE:

```
CALL LCRF (ICRFF, RANGES, RINDEX,
          L, NULL, EOF, ERROR)
```

ROUTINE: LCSF

FUNCTION: Reads one record from the CSF file and converts all significant data to an integer value from 1 to N

CALLING SEQUENCE:

```
CALL LCSF (ICSFF, RANGES, RINDEX,
          L, NULL, EOF, ERROR)
```

ROUTINE: LRAF

FUNCTION: Reads one record from the RAF file and converts all significant data to an integer value from 1 to N

CALLING SEQUENCE:

```
CALL LRAF (IRAFF, RANGES, RINDEX,
          L, NULL, EOF, ERROR)
```

ROUTINE: PROFIL

FUNCTION: Main routine of the PF program, produces the profile report for the project and file type specified

CALLING SEQUENCE: None

3.2.3.2 Write Output Report and Plot Files

These five routines write the output report and plot files.

ROUTINE: CENTTL

FUNCTION: Centers the character titles

CALLING SEQUENCE:

```
CALL CENTTL (NAMES,
             SBTTLS)
```

ROUTINE: HEADER

FUNCTION: Prints a one-line title for each report page that includes the date and project name

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PRJNAM, RPTITL)
```

ROUTINE: HSUMRY

FUNCTION: Prints a six-line header summary with data from the Phase Dates (HDR) and Estimated Statistics (EST) files

CALLING SEQUENCE:

```
CALL HSUMRY (IRPTF, PRJNAM)
```

ROUTINE: PRTDAT

FUNCTION: Prints report data

CALLING SEQUENCE:

```
CALL PRTDAT (BRKVAR, CATNAM, CATSIZ, IRPTF, K, KTOT,
             NCAT, PRJNAM, RANGES, RNGCHK, RPTITL,
             RPTNAM, STEPS, VARNUM)
```

ROUTINE: WRTPLT

FUNCTION: Writes profile statistics to a temporary file in preparation for plotting

CALLING SEQUENCE:

```
CALL WRTPLT (BRKVAR, CATNAM, CATSIZ, FILTYP, K, KTOT,
             MAKPLT, NCAT, PRJNAM, RPTITL, STEPS, VARNUM)
```

3.2.3.3 Obtain Data From Terminal or External File

These four routines obtain information from a user's response to a terminal prompt or from an external file.

ROUTINE: GETCOL

FUNCTION: Reads the PF description file to obtain descriptions of fields and categories for the selected profile report

CALLING SEQUENCE:

```
CALL GETCOL (BRKV, COLFIL,
             BRKVAR, CATNAM, CATSIZ, MAKPLT, NCAT, RANGES,
             RINDEX, RNGCHK, RPTITL, STEPS, VARNUM, ERROR)
```

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR
             FIELD)
```

ROUTINE: GETOPT

FUNCTION: Obtains the project name and user options from the terminal

CALLING SEQUENCE:

```
CALL GETOPT (TERMNL,
             BRKV, COLFIL, DBFILE, FILTYP, PRJNAM, RPTNAM,
             EOF, ERROR)
```

ROUTINE: HELP

FUNCTION: Prints help information to the terminal

CALLING SEQUENCE:

```
CALL HELP
```

ROUTINE:: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

3.2.3.4 File Open and Read Routines

These eight routines either open an indexed file or read records from an indexed file.

ROUTINE: DOPEN2

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL DOPEN2 (IFILE, FILNAM,
             FOUND, ERROR)
```

ROUTINE: RDCIF

FUNCTION: Reads one record from the CIF and converts all data to internal format

CALLING SEQUENCE:

```
CALL RDCIF (ICIFF,
            PROJNO, CNAME, ICODE, PANV, MODFUN, SYSFUN,
            ORIGIN, NEXEC, NLINES, NCOMNT, IETA1, IETA2,
            NETA1, NETA2, NIOVAR, MCCABE, NFUNCT, NIO,
            NASGN, NCALL, NFMT, STATUS, EOF, ERROR)
```

ROUTINE: RDCRF

FUNCTION: Reads one record from the CRF file and converts the data to internal format

CALLING SEQUENCE:

```
CALL RDCRF (ICRFF,
            FORMNO, PROJNO, PROGNO, FDATE, NCH, NEXAM,
            OVER1, DATDET, DATBEG, EFFORT, CHTYPE, CHCOMP,
            ERRTYP, ERRIN, DATERR, LGCERR, ACTVTY, ISOLTM,
            PATCH, RELOLD, RELNO, RELDAT, CMTREA, CMTDES,
            CMTGEN, STATUS, EOF, ERROR)
```

ROUTINE: RDCSF

FUNCTION: Reads one record from the CSF file

CALLING SEQUENCE:

```
CALL RDCSF (ICSFF,
            FORMNO, PROJNO, PROGNO, PROGI, FDATE, FSTAGE,
            COMPCO, PRECIS, CMPLEX, SWTYPE, PASGN, PCNTL,
            POTHER, STATWO, STMT, BTSIZE, INDEP, RELSW,
            ADDTYP, NCALLD, X1, NCALNG, X2, NSHR, X3,
            NDESC, X4, LANG1, PLANG1, LANG2, PLANG2, DES,
            CONSTR, DESRUN, CODRUN, TSTRUN, DESTIM,
            CODTIM, TSTTIM, DESEFF, CODEFF, TSTEFF,
            DESDAT, CODDAT, TSTDAT, DESCR, CALLD, CALNG,
            SHR, AFFECT, OTH, NAMCON, CMT1, CMT2, ISTAT,
            EOF, ERROR)
```

ROUTINE: RDEST

FUNCTION: Reads one record from the EST file and converts all data to internal format

CALLING SEQUENCE:

```
CALL RDEST (IESTF,
           NAME, PROJ, NCOMP, MODDEL, MODNEW, MODMOD,
           NRUNS, NCHANG, PAGDOC, LINDEL, LINNEW,
           LINMOD, TOTEXT, NEWEXT, MODEXT, PROGHR,
           MGMTHR, OTHRHR, HR95, HR75, OTHCMP, STATUS,
           ACTIVE, PRJCAT, FOUND, ERROR)
```

ROUTINE: RDHDRX

FUNCTION: Reads the HDR file and returns the phase dates for a given project

CALLING SEQUENCE:

```
CALL RDHDRX (IHDRF, PROJCT,
             DRANG1, DRANG2, FOUND)
```

ROUTINE: RDHDR1

FUNCTION: Reads one record from the HDR file and converts all data to internal format

CALLING SEQUENCE:

```
CALL RDHDR1 (IHDRF, PRJNAM,
             PROJ, DEVCMP, TARG, ALIEN, REQ1, REQ2, DES1,
             DES2, CODE1, CODE2, SYS1, SYS2, ACC1, ACC2,
             CLEAN1, CLEAN2, MAINT1, MAINT2, STATUS,
             FOUND, ERROR)
```

ROUTINE: RDRAF

FUNCTION: Reads one record from the RAF file

CALLING SEQUENCE:

```
CALL RDRAF (IRAFF,
            FORMNO, SEQNO, PROJNO, PROGNO, RDATE, MACHIN,
            INTERA, PURPOS, NCOMP, COMPCO, FIRST, METOBJ,
            RESULT, COMENT, ISTAT, EOF, ERROR)
```

3.2.3.5 Routines for String Movement, Comparison, or Conversion

These eight routines deal with string movement, comparison, or conversion.

ROUTINE: BLANK

FUNCTION: Initializes an array to blanks

CALLING SEQUENCE:

```
CALL BLANK (ARRAY, NUM)
```

ROUTINE: CHRINT

FUNCTION: Converts the given string to integer in I*2 format

CALLING SEQUENCE:

```
CALL CHRINT (CHARS, NCHAR,
             I2NUM, ERROR)
```

ROUTINE: CNVRNG

FUNCTION: Converts the given range to character format

CALLING SEQUENCE:

```
CALL CNVRNG (IBRK, IRNG, RANGES,
             SUBTTL)
```

ROUTINE: MATCHS (LOGICAL FUNCTION)

FUNCTION: Determines whether two input strings are the same

CALLING SEQUENCE:

```
MATCHS (ARRAY1, ARRAY2, NBYTES)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: SQEEZ

FUNCTION: Removes blanks from a character string

CALLING SEQUENCE:

```
CALL SQEEZ (IN, NSIZE,
            NONBL, OUT)
```

ROUTINE: WHERE

FUNCTION: Finds the location of the given character in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
            LOC, FOUND)
```

3.2.3.6 Variable Description

The variables in the calling sequences of major PF routines are described below.

Name	Type	Description
BRKV	I*2	Item number of variable desired as breakdown variable
BRKVAR	I*2	Number of categories in PF description file for the breakdown variable

Name	Type	Description
CATNAM(25,20)	L*1	Array of field names for each field in PF description file
CATSIZ(20)	I*2	Number of categories for each field
COLFIL(27)	L*1	Name of PF description file
DBFILE(27)	L*1	Data base file name to be read
EOF	L*1	End-of-file flag
EOFTTY	L*1	Flag for end of file on terminal
ERROR	L*1	Error flag
EXTFIL	I*2	FORTRAN unit number for external file to be read
FIELD(FLDLEN)	L*1	Field to be obtained
FILTYP	L*1	Character indicating type of report desired: I = CIF, H = CRF, M = CSF, A = RAF
FLDLEN	I*2	Length of field
IBRK	I*2	Index of boundary of category range to convert (from category description record)
ICIFF	I*2	FORTRAN unit number of the CIF
ICRFF	I*2	FORTRAN unit number of the CRF file
ICSFF	I*2	FORTRAN unit number of the CSF file
IDBF	I*2	Data base file unit number to be read
IRAFF	I*2	FORTRAN unit number of the RAF file
IRNG	I*2	Index of category range to convert (row number of category on report)
IRPTF	I*2	Profile report file unit number
K(9,8,20)	I*2	Data array containing all data for profile report except totals
KTOT(9)	I*2	Array of totals for total column on profile report
L(55)	I*2	Integer representation of each type of data

Name	Type	Description
MAKPLT(20)	L*1	Array of switches indicating whether a plot file is to be produced
NAMES(12,8)	L*1	Titles
NCAT	I*2	Number of fields in profile report
NULL	L*1	Flag indicating if this record is usable
PRJNAM(8)	L*1	Project name
RANGES(9,55)	I*2	Range boundaries for all fields identified with asterisks in column 5 of the PF description file
RINDEX(55)	I*2	Array used for sorting capability (not currently implemented)
RNGCHK(20)	L*1	Array of flags for each field indicating whether the categories for the field are ranges of values
RPTITL(40)	L*1	Report title
RPTNAM(27)	L*1	Report file name
SBTTLS(12,8)	L*1	Centered titles
STEPS(12,8,20)	L*1	Array of category names for each field
SUBTTL(12)	L*1	Array containing column titles
TERMNL	L*1	Flag indicating whether response is to be read from the terminal or an external file
TEXT(FLDLEN)	L*1	Prompt text string
VARNUM(20)	I*2	Item numbers for each field

3.2.4 TASK BUILD PROCEDURE

3.2.4.1 Command Procedures

The PF program can be generated from the source code by executing the command procedure PFGEN.CMD under UIC [204,6]. This command procedure references three command files--PFFPP.CMD, PFFOR.CMD, and PF.TKB--all under UIC [204,6]. Figure 3-5 is a listing of PFGEN.CMD, the command procedure

```
;                                                                    1
;   @PFGEN.CMD                                                       2
;                                                                    3
;   THIS COMMAND PROCEDURE PRECOMPILES, COMPILES, AND TASK BUILDS    4
;   THE PROFILE REPORT PROGRAM (PF).                                 5
;                                                                    6
;   PRECOMPILE ROUTINES WRITTEN IN STRUCTURED FORTRAN                7
;                                                                    8
@[204,6]PFFPP.CMD                                                    9
;                                                                   10
;                                                                   11
;   @PFFPP.CMD                                                      12
;                                                                   13
;   THIS COMMAND PROCEDURE PRECOMPILES ALL SOURCE CODES WRITTEN IN  14
;   STRUCTURED FORTRAN FOR THE PROFILE REPORT PROGRAM (PF).         15
;                                                                   16
;   ROUTINES WITH PREFIX PF                                         17
;                                                                   18
;FPP SY:[204,6]PFCENTTL                                             19
;FPP SY:[204,6]PFCNVRNG                                             20
;FPP SY:[204,6]PFGETCOL                                             21
;FPP SY:[204,6]PFGETDAT                                             22
;FPP SY:[204,6]PFGETOPT                                             23
;FPP SY:[204,6]PFHELP                                               24
;FPP SY:[204,6]PFLCIF                                               25
;FPP SY:[204,6]PFLCRF                                               26
;FPP SY:[204,6]PFLCSF                                               27
;FPP SY:[204,6]PFLRAF                                               28
;FPP SY:[204,6]PFPROFIL                                             29
;FPP SY:[204,6]PFPRTDAT                                             30
;FPP SY:[204,6]PFWRTPLT                                             31
;                                                                   32
;   ROUTINES WITH PREFIX UT                                         33
;                                                                   34
;FPP SY:[204,7]UTBLANK                                              35
;FPP SY:[204,7]UTCHRINT                                             36
;FPP SY:[204,7]UTDOPEN2                                             37
;FPP SY:[204,7]UTGETFLD                                             38
;FPP SY:[204,7]UTGETLEN                                             39
;FPP SY:[204,7]UTHEADER                                             40
;FPP SY:[204,7]UTHSUMRY                                             41
;FPP SY:[204,7]UTMATCHS                                             42
;FPP SY:[204,7]UTMOVE                                               43
;FPP SY:[204,7]UTNAME3                                              44
;FPP SY:[204,7]UTRDCIF                                              45
;FPP SY:[204,7]UTRDCRF                                              46
;FPP SY:[204,7]UTRDCSF                                              47
;FPP SY:[204,7]UTRDEST                                              48
;FPP SY:[204,7]UTRDHDRX                                             49
;FPP SY:[204,7]UTRDHDR1                                             50
;FPP SY:[204,7]UTRDRAF                                              51
;FPP SY:[204,7]UTSQEEZ                                              52
;FPP SY:[204,7]UTWHERE                                              53
;                                                                   54
;   COMPILE FORTRAN ROUTINES                                        55
```

Figure 3-5. PF Task Generation Command Procedure (PFGEN.CMD) (1 of 2)

```
;                                                                              56
@[204,6]PFFOR.CMD                                                              57
;                                                                              58
;    @PFFOR.CMD                                                                59
;                                                                              60
;    THIS COMMAND PROCEDURE COMPILES ALL FORTRAN ROUTINES FOR THE PROFILE       61
;    REPORT PROGRAM (PF).                                                      62
;                                                                              63
;    ROUTINES WITH PREFIX PF                                                   64
;                                                                              65
;FOR/F4P/OBJECT:[204,6]PFCENTTL [204,6]PFCENTTL                                66
;FOR/F4P/OBJECT:[204,6]PFCNVRNG [204,6]PFCNVRNG                                67
;FOR/F4P/OBJECT:[204,6]PFGETCOL [204,6]PFGETCOL                                68
;FOR/F4P/OBJECT:[204,6]PFGETDAT [204,6]PFGETDAT                                69
;FOR/F4P/OBJECT:[204,6]PFGETOPT [204,6]PFGETOPT                                70
;FOR/F4P/OBJECT:[204,6]PFHELP   [204,6]PFHELP                                  71
;FOR/F4P/OBJECT:[204,6]PFLCIF   [204,6]PFLCIF                                  72
;FOR/F4P/OBJECT:[204,6]PFLCRF   [204,6]PFLCRF                                  73
;FOR/F4P/OBJECT:[204,6]PFLCSF   [204,6]PFLCSF                                  74
;FOR/F4P/OBJECT:[204,6]PFLRAF   [204,6]PFLRAF                                  75
;FOR/F4P/OBJECT:[204,6]PFPROFIL [204,6]PFPROFIL                                76
;FOR/F4P/OBJECT:[204,6]PFPRTDAT [204,6]PFPRTDAT                                77
;FOR/F4P/OBJECT:[204,6]PFWRTPLT [204,6]PFWRTPLT                                78
;                                                                             79
;    ROUTINES WITH PREFIX UT                                                   80
;                                                                              81
;FOR/F4P/OBJECT:[204,7]UTBLANK  [204,7]UTBLANK                                 82
;FOR/F4P/OBJECT:[204,7]UTCHRINT [204,7]UTCHRINT                                83
;FOR/F4P/OBJECT:[204,7]UTDOPEN2 [204,7]UTDOPEN2                                84
;FOR/F4P/OBJECT:[204,7]UTGETFLD [204,7]UTGETFLD                                85
;FOR/F4P/OBJECT:[204,7]UTGETLEN [204,7]UTGETLEN                                86
;FOR/F4P/OBJECT:[204,7]UTHEADER [204,7]UTHEADER                                87
;FOR/F4P/OBJECT:[204,7]UTHSUMRY [204,7]UTHSUMRY                                88
;FOR/F4P/OBJECT:[204,7]UTMATCHS [204,7]UTMATCHS                                89
;FOR/F4P/OBJECT:[204,7]UTMOVE   [204,7]UTMOVE                                  90
;FOR/F4P/OBJECT:[204,7]UTNAME3  [204,7]UTNAME3                                 91
;FOR/F4P/OBJECT:[204,7]UTRDCIF  [204,7]UTRDCIF                                 92
;FOR/F4P/OBJECT:[204,7]UTRDCRF  [204,7]UTRDCRF                                 93
;FOR/F4P/OBJECT:[204,7]UTRDCSF  [204,7]UTRDCSF                                 94
;FOR/F4P/OBJECT:[204,7]UTRDEST  [204,7]UTRDEST                                 95
;FOR/F4P/OBJECT:[204,7]UTRDHDRX [204,7]UTRDHDRX                                96
;FOR/F4P/OBJECT:[204,7]UTRDHDR1 [204,7]UTRDHDR1                                97
;FOR/F4P/OBJECT:[204,7]UTRDRAF  [204,7]UTRDRAF                                 98
;FOR/F4P/OBJECT:[204,7]UTSQEEZ  [204,7]UTSQEEZ                                 99
;FOR/F4P/OBJECT:[204,7]UTWHERE  [204,7]UTWHERE                                100
;                                                                            101
;    GENERATE THE TASK IMAGE                                                 102
;                                                                            103
TKB @[204,6]PF.TKB                                                           104
;                                                                            105
;    @PF.TKB                                                                 106
;                                                                            107
;    COMMAND PROCEDURE TO BUILD THE TASK IMAGE F4P THE PROFILE REPORT        108
;    PROGRAM (PF)                                                            109
;                                                                            110
;[204,5]PF/FU=[204,6]PF/MP                                                   111
;UNITS=20                                                                    112
;MAXBUF=250                                                                  113
;//                                                                         114
```

Figure 3-5. PF Task Generation Command Procedure (PFGEN.CMD) (2 of 2)

to precompile, compile, and task build the PF program. The PF program is generated by entering the following command:

```
@[204,6]PFGEN
```

3.2.4.2 Overlay Structure

The PF program is overlaid to reduce the memory space requirement. Figure 3-6 is a listing of the Overlay Descriptor Language file, [204,6]PF.ODL, needed to build the PF program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay. In addition, the Record Management Service (RMS) Indexed Access Programs Library (RMSIAC) is also needed in the overlay. The name of the library is UFRMSIAC.OLB under UIC [204,7]. It contains FORTRAN routines used for accessing RMS indexed files.

```
;                                                                              1
;   @PF.ODL                                                                    2
;                                                                              3
;   THE OVERLAY DEFINITION FOR THE PROFILE REPORT PROGRAM (PF)                 4
;                                                                              5
          .ROOT $TREE1,OTSALL,RMSALL                                           6
$TREE1:   .FCTR $ROOT-RMSROT-OTSROT-*($OPT,$COL,$DAT,$PRT,$PLT)                7
$ROOT:    .FCTR [204,6]PFPROFIL-[204,7]UTMOVE  -[204,7]UTDOPEN2-$ROOT3         8
$ROOT3:   .FCTR [204,6]PFGETDAT-[204,7]UTSQEEZ -[204,7]UTWHERE -$ROOT5         9
$ROOT5:   .FCTR [204,7]UFRMSIAC/LB                                            10
;                                                                             11
$OPT:     .FCTR [204,6]PFGETOPT-[204,7]UTNAME3 -[204,6]PFHELP-$OPT2           12
$OPT2:    .FCTR [204,7]UTGETFLD-[204,7]UTCHRINT                               13
;                                                                             14
$COL:     .FCTR [204,6]PFGETCOL-[204,6]PFCNVRNG                               15
;                                                                             16
$DAT:     .FCTR ($CIF,$CRF,$CSF,$RAF)                                         17
$CIF:     .FCTR [204,6]PFLCIF  -[204,7]UTRDCIF-[204,7]UFRMSIAC/LB             18
$CRF:     .FCTR [204,6]PFLCRF  -[204,7]UTRDCRF-[204,7]UFRMSIAC/LB             19
$CSF:     .FCTR [204,6]PFLCSF  -[204,7]UTRDCSF-[204,7]UFRMSIAC/LB             20
$RAF:     .FCTR [204,6]PFLRAF  -[204,7]UTRDRAF-[204,7]UFRMSIAC/LB             21
;                                                                             22
$PRT:     .FCTR [204,6]PFPRTDAT-[204,7]UTHEADER-[204,7]UTHSUMRY-$OUT2         23
$OUT2:    .FCTR [204,7]UTRDHDRX-[204,7]UTRDHDR1-[204,7]UTRDEST -$OUT3         24
$OUT3:    .FCTR [204,6]PFCENTTL-[204,7]UFRMSIAC/LB                            25
;                                                                             26
$PLT:     .FCTR [204,6]PFWRTPLT-[204,7]UTNAME3                                27
;                                                                             28
;                                                                             29
@LB:[1,1]RMS11M                                                               30
@LB:[1,1]RMS12X                                                               31
          .END                                                                32
```

Figure 3-6. PF Program Overlay Descriptor Language File (PF.ODL)

3.3 RESOURCE UTILIZATON REPORT PROGRAM (RU)

3.3.1 INTRODUCTION

The Resource Utilization Report Program (RU) produces a report of manpower and computer resource data subdivided by phase for a given project. The resource data used are obtained from the Component Status Report (CSR) file and the Resource Summary Form (RSF) file for the given project.

3.3.2 PROGRAM STRUCTURE

3.3.2.1 Files Accessed

The RU program accesses five input files and five output files as described below.

Input File Name	Description
[204,6]RU.NL	A sequential file containing the key input parameters (a user-defined RU input parameters file under the UIC may be provided instead)
[204,1]EST.HDR	Estimated Statistics (EST) file
[204,1]HEADER.HDR	Phase Dates (HDR) file
[204,1]<PRJNAM>.CSR	CSR file for the given project
[204,1]<PRJNAM>.RSF	RSF file for the given project

Output File Name	Description
<PRJNAM>.RU	File containing the RU report for the given project
<PRJNAM>.1RU	First plot file for the given project, containing data from the RSF file (subdivided by phase)
<PRJNAM>.2RU	First plot file for the given project, containing data from the CSR file (subdivided by phase)
<PRJNAM>.3RU	Second plot file for the given project, containing data from the RSF file (subdivided by manpower category)

Output File Name	Description
<PRJNAM>.4RU	Second plot file for the given project, containing data from the CSR file (subdivided by manpower category)

In these file names, <PRJNAM> is the name of the project selected by the user. The four plot output files are intended for use by the Pie Chart Plotting Program, which is not currently implemented.

3.3.2.2 Baseline Diagram

Figure 3-7 is the baseline diagram for the RU program. The RU routine is the main driver. It reads the RU input parameters file, the EST file, the HDR file, the RSF file, and the CSR file and prints the resource utilization report. RU loops through the above process until a^Z (control Z) is returned by the user in response to a prompt.

3.3.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the RU program are grouped here by functions. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of the variables begins a new line. The calling sequence variables for the major RU routines are described in Section 3.3.3.6. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the RU program also uses the following system routines: DATE, ERRSNS, and TIME.

3.3.3.1 Process Data and Compute Statistics

These seven major routines obtain data from a given CSR or RSF file and compute statistics for the RU report.

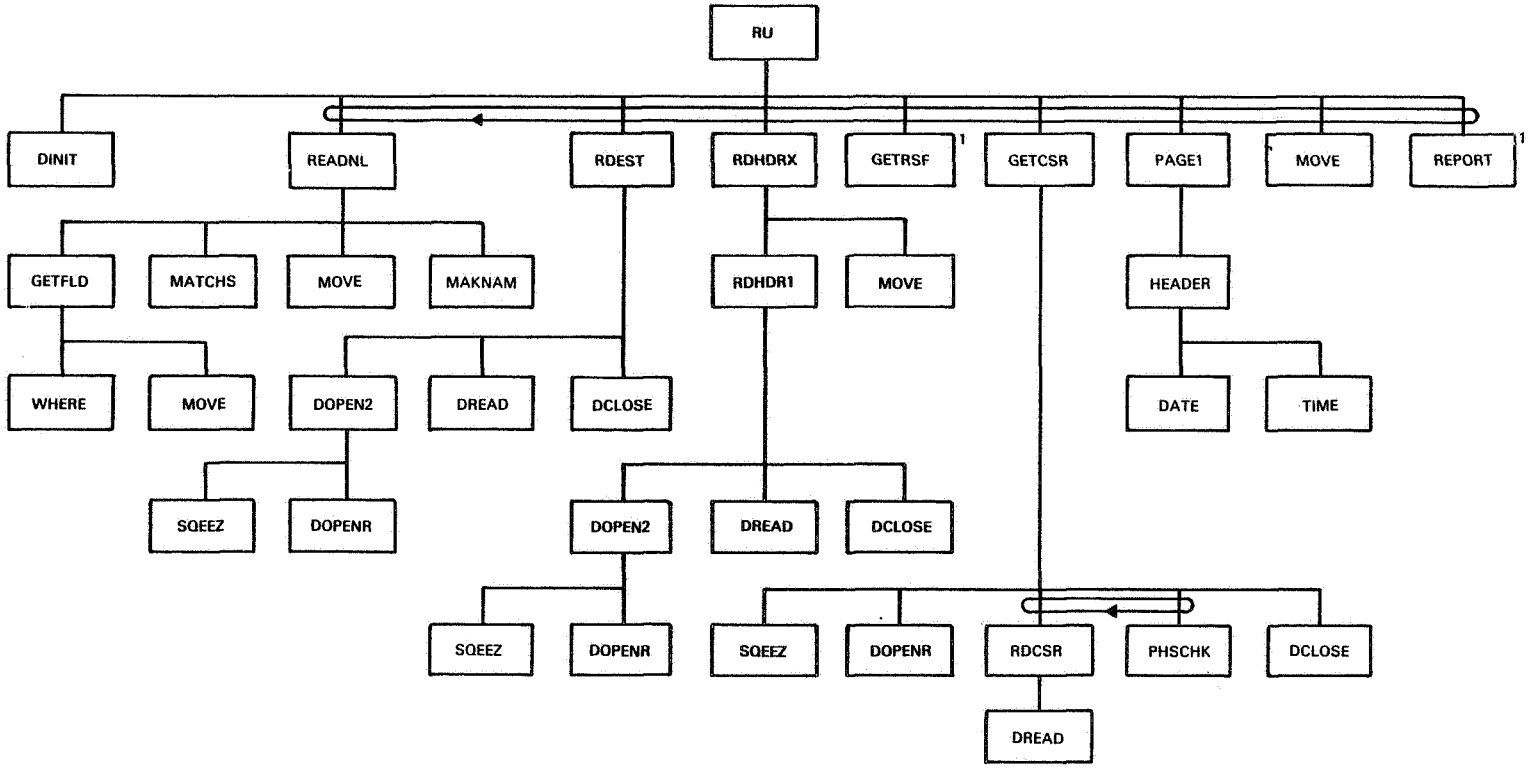


Figure 3-7. Baseline Diagram for the Resource Utilization Report Program (RU) (1 of 2)

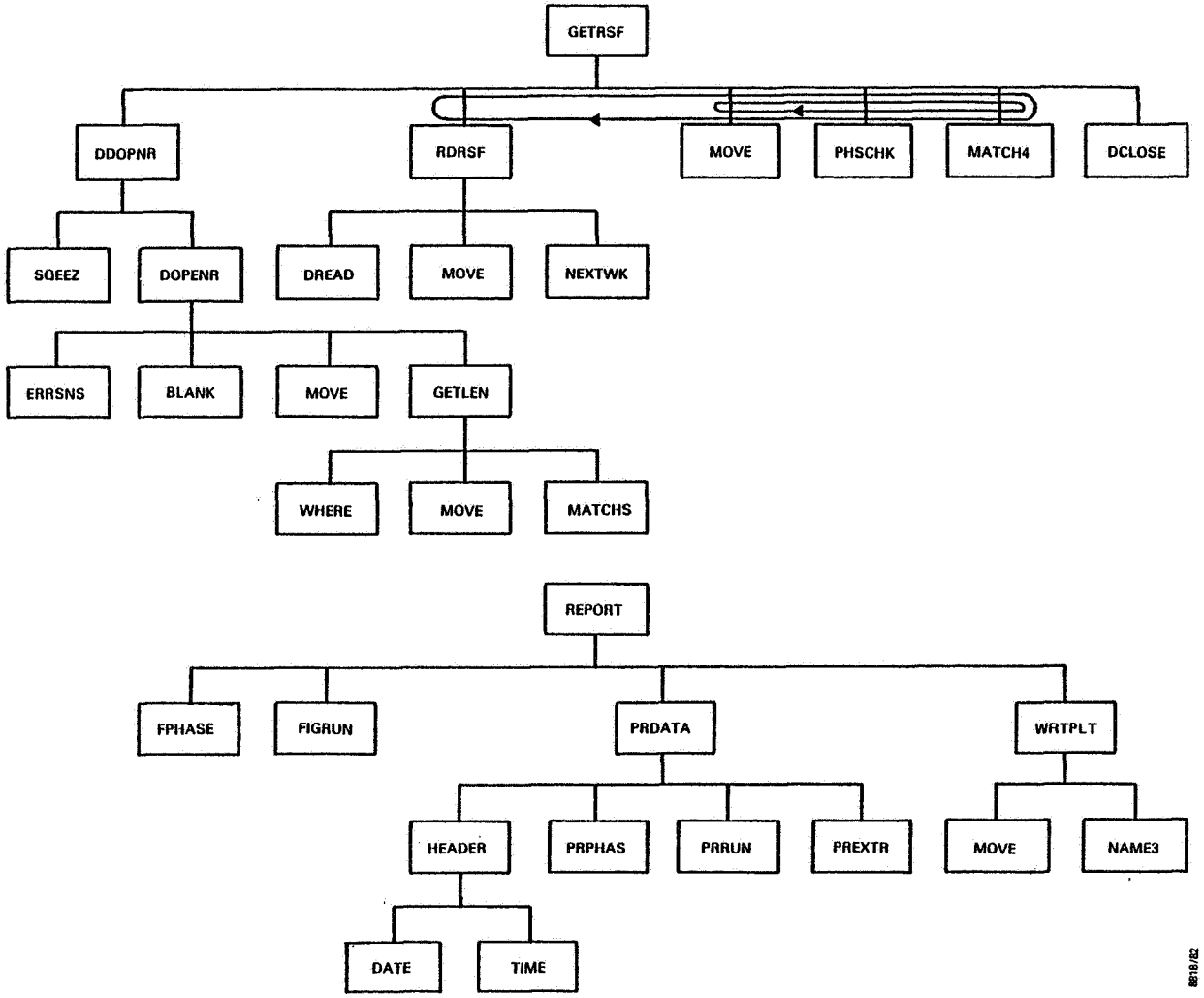


Figure 3-7. Baseline Diagram for the Resource Utilization Report Program (RU) (2 of 2)

ROUTINE: FIGRUN

FUNCTION: Computes all data used in the second section of the body of the RU report (computer usage, source code size, and change data)

CALLING SEQUENCE:

```
CALL FIGRUN (XCOST, LNMULT, XMMM, XMWTMM, HR75, HR95,
             COMDEL, COMNEW, OLDFAC, XPMM, XPWTMM, RUNS,
             XSMM, XSWTMM, LINDEL, LINNEW, T95T75, NCHANG,
             COSPER, EQU75, E75PER, LP, LPM, LPMS, WTLP,
             WTLPM, WTLPMS, H75PER, H95PER, NCOMP, RUNPER,
             SLINES, CHGPER)
```

ROUTINE: FPHASE

FUNCTION: Computes all necessary phase data

CALLING SEQUENCE:

```
CALL FPHASE (COSTHR, HRMON, MGHR, MGWT, NWEEKS, PROGHR,
             PRWT, SVHR, SVWT,
             MHR, MMM, MPCT, MWTHR, MWTMM, MWTPCT, MCOST,
             MPHSPC, PHR, PMM, PPCT, PWTHR, PWTMM, PWTPCT,
             PCOST, PPHSPC, SHR, SMM, SPCT, SWTHR, SWTMM,
             SWTPCT, SCOST, SPHSPC, THR, TMM, TPCT, TWTHR,
             TWTMM, TWTPCT, TCOST, TPHSPC, WEEKPC)
```

ROUTINE: GETCSR

FUNCTION: Obtains programmer hour totals by phase from the CSR file

CALLING SEQUENCE:

```
CALL GETCSR (CSRFIL, CSRNAM, DRANG1, DRANG2,
             CPRGHR)
```

ROUTINE: GETRSF

FUNCTION: Reads all of the RSF file and accumulates programmer, management, and services hours for each phase

CALLING SEQUENCE:

```
CALL GETRSF (DRANG1, DRANG2, IRSFF, RSFFIL,
             MGHR, RSFPHR, SVHR, ERROR)
```

ROUTINE: NEXTWK

FUNCTION: Computes data one week after the given date and returns it in YYMMDD format

CALLING SEQUENCE:

```
CALL NEXTWK (DATE,
             D)
```

ROUTINE: REPORT

FUNCTION: Given the key input parameters and RSF or CSR data, computes and prints percentages and totals

CALLING SEQUENCE:

```
CALL REPORT (IRPTF, PAGENO, PHRASE, COSTHR, HRMON,
             LNMULT, MGWT, OLDFAC, PRWT, SVWT, T95T75,
             DRANG1, DRANG2, MGHR, HR75, HR95, NWEEKS,
             PHRANG, PROGHR, PROJCT, RUNS, SVHR, COMDEL,
             COMNEW, LINDEL, LINNEW, NCHANG, NDATWK, TURN)
```

ROUTINE: RU

FUNCTION: Main routine of the RU program, reads the RSF and CSR files and prints the RU report

CALLING SEQUENCE: None

3.3.3.2 Write the RU Report and Plot Files

These seven routines write the RU output report and plot files.

ROUTINE: HEADER

FUNCTION: Prints a one-line title for each report page that includes the date, project name, and page number

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PROJCT, PAGENO)
```

ROUTINE: PAGE1

FUNCTION: Prints abbreviations and notes and key input parameters on the first page of the RU report

CALLING SEQUENCE:

```
CALL PAGE1 (IRPTF, PAGENO, COSTHR, COSTMM, CSTFIL, HRMON,
           HRYR, LNMULT, MGWT, NLFIL, OLDFAC, PRWT,
           PROJCT, RPTFIL, RSFFIL, SVWT, T95T75)
```

ROUTINE: PRDATA

FUNCTION: Prints Sections 1 and 2 of the body (pages 2 and 3) of the RU report

CALLING SEQUENCE:

```
CALL PRDATA (IRPTF, PAGENO, PHRASE, PROJCT, DRANG1,
             DRANG2, MGCST, MGRSF, NWEEKS, PHRANG, PRGCST,
             PRGRSF, MHR, MMM, MPCT, MWTHR, MWTMM,
             MWTPCT, MCOST, MPHSPC, PHR, PMM, PPCT,
             PWTHR, PWTMM, PWTPCT, PCOST, PPHSPC, SHR,
             SMM, SPCT, SWTHR, SWTMM, SWTPCT, SCOST,
             SPHSPC, THR, TMM, TPCT, TWTHR, TWTMM, TWTPCT,
             TCOST, TPHSPC, WEEKPC, COST, COSPER, EQU75,
             E75PER, LNMULT, LP, LPM, LPMS, WTLP, WTLPM,
             WTLPMS, HR75, H75PER, HR95, H95PER, NCOMP,
             RUNS, RUNPER, SLINES, NONCOM, COSPM, COSPMS,
             NCHANG, CHGPER)
```

ROUTINE: PREXTR

FUNCTION: Prints the third section of the body of the RU report

CALLING SEQUENCE:

```
CALL PREXTR (IRPTF, NONCOM, COSPM, COSPMS)
```

ROUTINE: PRPHAS

FUNCTION: Prints all data that have been processed according to phase

CALLING SEQUENCE:

```
CALL PRPHAS (IRPTF, DRANG1, DRANG2, MGCST, MGRSF,
             NWEEKS, PHRANG, PRGCST, PRGRSF, MHR, MMM,
             MPCT, MWTHR, MWTMM, MWTPCT, MCOST, MPHSPC,
             PHR, PMM, PPCT, PWTHR, PWTMM, PWTPCT, PCOST,
             PPHSPC, SHR, SMM, SPCT, SWTHR, SWTMM,
             SWTPCT, SCOST, SPHSPC, THR, TMM, TPCT,
             TWTHR, TWTMM, TWTPCT, TCOST, TPHSPC, WEEKPC)
```

ROUTINE: PRRUN

FUNCTION: Prints various data related to source lines, computer usage, and changes

CALLING SEQUENCE:

```
CALL PRRUN (IRPTF, COST, COSPER, EQU75, E75PER, LNMULT,
            LP, LPM, LPMS, WTLP, WTLPM, WTLPMS, HR75,
            H75PER, HR95, H95PER, NCOMP, RUNS, RUNPER,
            SLINES, NCHANG, CHGPER)
```

ROUTINE: WRTPLT

FUNCTION: Writes data to two intermediate files in preparation for pie chart plotting

CALLING SEQUENCE:

```
CALL WRTPLT (MGMTHR, PRJNAM, PROGHR, SERVHR, TURN)
```

3.3.3.3 Obtain Data From Terminal or External File

These four routines obtain information from a user's response to a terminal prompt or from an external file.

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR
             FIELD)
```

ROUTINE: MAKNAM

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL MAKNAM (DISK, UIC, NAME, EXTENS,
             DSN)
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

ROUTINE: READNL

FUNCTION: Reads the RU input parameters file

CALLING SEQUENCE:

```
CALL READNL (INLF,
             TERMNL,
             COSTHR, COSTMM, CSTFIL, HRMON, HRYR, LNMULT,
             MGWI, NLFIL, OLDFAC, PROJ, PRWT, RPTFIL,
             RSFFIL, SVWT, T95T75, EOF, ERROR)
```

3.3.3.4 File Open and Read Routines

These seven routines either open an indexed file or read records from an indexed file.

ROUTINE: DDOPNR

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
    CALL DDOPNR (IFILE, FILNAM,
                 FOUND, ERROR)
```

ROUTINE: DOPEN2

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
    CALL DOPEN2 (IFILE, FILNAM,
                 FOUND, ERROR)
```

ROUTINE: RDCSR

FUNCTION: Reads one record from the CSR file

CALLING SEQUENCE:

```
    CALL RDCSR (CSRFIL,
                FORMNO, SEQNO, PROJNO, PROGNO, FDATE, COMPCO,
                TIMES, OTHNAM, OTHOUR, ISTAT, PHASE, EOF,
                ERROR)
```

ROUTINE: RDEST

FUNCTION: Reads one record from the EST file and converts all data to internal format

CALLING SEQUENCE:

```
    CALL RDEST (IESTF, NAME,
                PROJ, NCOMP, MODDEL, MODNEW, MODMOD, NRUNS,
                NCHANG, PAGDOC, LINDEL, LINNEW, LINMOD,
                TOTEXT, NEWEXT, MODEXT, PROGHR, MGMTHR,
                OTHRHR, HR95, HR75, OTHCMP, STATUS, ACTIVE,
                PRJCAT, FOUND, ERROR)
```

ROUTINE: RDHDRX

FUNCTION: Reads the HDR file and returns the phase dates for a given project

CALLING SEQUENCE:

```
CALL RDHDRX (IHDRF, PROJCT,
             DRANG1, DRANG2, FOUND)
```

ROUTINE: RDHDR1

FUNCTION: Reads one record from the HDR file and converts all data to internal format

CALLING SEQUENCE:

```
CALL RDHDR1 (IHDRF, PRJNAM,
             PROJ, DEVCMP, TARG, ALIEN, REQ1, REQ2, DES1,
             DES2, CODE1, CODE2, SYS1, SYS2, ACC1, ACC2,
             CLEAN1, CLEAN2, MAINT1, MAINT2, STATUS,
             FOUND, ERROR)
```

ROUTINE: RDRSF

FUNCTION: Reads one record on the RSF file and returns all data on that record plus an array of week dates for each resource entry on the record

CALLING SEQUENCE:

```
CALL RDRSF (RSFFIL,
            FORMNO, SEQNO, PROJNO, RESCOD, RESID, FDATE,
            PCMGMT, WKDATE, NRUNS, TIMES, STATUS, PHASE,
            LASTWK, EOF, ERROR)
```

3.3.3.5 Routines for String Movement or Comparison

These eight routines deal with string movement or comparison.

ROUTINE: MATCHS (LOGICAL FUNCTION)

FUNCTION: Determines whether the two input strings are the same

CALLING SEQUENCE:

```
MATCHS (ARRAY1, ARRAY2, NBYTES)
```

ROUTINE: MATCH4 (LOGICAL FUNCTION)

FUNCTION: Determines whether a given number is in a given array

CALLING SEQUENCE:

```
MATCH4 (N, IARRAY, NARRAY)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: PHSCHK

FUNCTION: Determines if the given date is within the given date range

CALLING SEQUENCE:

```
CALL PHSCHK (FDATE, DRANG1, DRANG2,
             PHNUM, INPHAS)
```

ROUTINE: SQEEZ

FUNCTION: Removes blanks from a character string

CALLING SEQUENCE:

```
CALL SQEEZ (IN, NSIZE,
            NONBL, OUT)
```

ROUTINE: WHERE

FUNCTION: Finds the location of the given character in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
            LOC, FOUND)
```

3.3.3.6 Variable Description

The variables in the calling sequences of major RU routines are described below.

Name	Type	Description
CHGPER(3)	R*4	Number of changes per 1000 lines per type (new, delivered, adjusted)
COMDEL	I*2	Number of components delivered
COMNEW	I*2	Number of new components
COSPER(3)	R*4	Cost per type (new, delivered, adjusted)
COSPM	R*4	Cost per person-month using programmer and management time only
COSPMS	R*4	Cost per person-month using programmer, management, and services time
COST	R*4	Total cost based on weighted hours
COSTHR	R*4	Cost per hour
COSTMM	R*4	Cost per person-month
CPRGHR(6)	R*4	Total hours spent in each phase from CSR file record
CSRFIL	I*2	FORTRAN unit number for CSR file
CSRNAM(25)	L*1	CSR file name
CSTFIL(25)	L*1	CSR file name
DRANG1(3,6)	I*2	Phase start dates
DRANG2(3,6)	I*2	Phase end dates
EOF	L*1	Terminal EOF flag
EQU75	R*4	IBM S/360-95 plus S/360-75 computer time in equivalent S/360-75 time

Name	Type	Description
ERROR	L*1	Error flag
E75PER(3)	R*4	Equivalent S/360-75 computer time per type (new, delivered, adjusted)
HRMON	R*4	Hours per month
HRYR	R*4	Hours per year
HR75	R*4	S/360-75 computer time in hours
HR95	R*4	S/360-95 computer time in hours
H75PER(3)	R*4	S/360-75 computer time per type (new, delivered, adjusted)
H95PER(3)	R*4	S/360-95 computer time per type (new, delivered, adjusted)
INLF	I*2	FORTRAN unit number for RU input parameters file
IRPTF	I*2	FORTRAN unit number for RU output report file
IRSFF	I*2	FORTRAN unit number for RSF file
LINDEL	I*2	Number of delivered source lines (in thousands)
LINNEW	I*2	Number of new source lines (in thousands)
LNMULT	I*2	Source lines multiple used in computing statistics
LP(3)	I*2	Source lines produced per person-month using programmer time only
LPM(3)	I*2	Source lines produced per person-month using programmer and management time
LPMS(3)	I*2	Source lines produced per person-month using programmer, management, and services time
MCOST(6)	R*4	Weighted management cost
MGCST(6)	I*2	Number of CSR forms with management data
MGHR(6)	R*4	Management hours from the RSFs
MGMTHR(6)	I*2	Management hours by phase
MGRSF(6)	I*2	Number of RSFs with management data
MGWT	R*4	Management weight

Name	Type	Description
MHR(6)	I*2	Management hours by phase
MMM(6)	R*4	Management hours in person-months by phase
MPCT(6)	I*2	Percent of management hours in each phase
MPHSPC(6)	I*2	Percent of weighted management cost for each phase
MWTHR(6)	I*2	Weighted management hours by phase
MWTMM(6)	R*4	Weighted management hours in person-months by phase
MWTPCT(6)	I*2	Percent of weighted management hours of a phase
NCHANG	I*2	Number of changes
NCOMP(4)	I*2	Number of components by type (new, delivered, adjusted, old)
NDATWK(6)	I*2	Number of weeks with data in the phase
NLFIL(25)	L*1	RU input parameters file name
NONCOM	I*2	Source lines excluding comments
NWEEKS(6)	I*2	Number of weeks in phase
OLDFAC	R*4	Factor used to compute adjusted lines of code from old and new figures
PAGENO	I*2	Page number on report
PCOST(6)	R*4	Weighted programmer cost by phase
PHR(6)	I*2	Programmer hours by phase
PHRANG(6,2)	I*2	Number range of phases
PHRASE(50)	L*1	Title of RU report
PMM(6)	R*4	Programmer hours in person-months by phase
PPCT(6)	I*2	Percent of programmer hours in each phase
PPHSPC(6)	I*2	Percent of weighted programmer cost for each phase
PRGCST(6)	I*2	Number of CSR forms with programmer data
PRGRSF(6)	I*2	Number of RSFs with programmer data

Name	Type	Description
PRJNAM(8)	L*1	Project name
PROGHR(6)	I*2	Programmer hours by phase
PROJ(8)	L*1	Project name
PROJCT(8)	L*1	Project name from RSF file
PRWT	R*4	Programmer weight
PWTHR(6)	I*2	Weighted programmer hours by phase
PWTMM(6)	R*4	Weighted programmer hours in person-months by phase
PWTPCT(6)	I*2	Percent of weighted programmer hours of a phase
RPTFIL(25)	L*1	RU report file name
RSFFIL(25)	L*1	RSF file name
RSFPHR(6)	R*4	Programmer hours for each phase
RUNPER(3)	R*4	Number of runs per type (new, delivered, adjusted)
RUNS	I*2	Total number of runs
SCOST(6)	R*4	Weighted services cost for each phase
SERVHR(6)	I*2	Services hours by phase
SHR(6)	I*2	Services hours by phase
SLINES(4)	I*2	Number of source lines (in thousands) (new, delivered, adjusted, old)
SMM(6)	R*4	Services hours in person-months by phase
SPCT(6)	I*2	Percent of services hours in each phase
SPHSPC(6)	I*2	Percent of weighted services cost for each phase
SVHR(6)	R*4	Services hours for each phase
SVWT	R*4	Services weight
SWTHR(6)	I*2	Weighted services hours by phase
SWTMM(6)	R*4	Weighted services hours in person-months by phase
SWTPCT(6)	I*2	Percent of weighted services hours for each phase
TCOST(6)	R*4	Weighted total cost for each phase

Name	Type	Description
TERMNL	L*1	Flag indicating whether terminal or external file is to be read
THR(6)	I*2	Total hours by phase
TMM(6)	R*4	Total hours in person-months by phase
TPCT(6)	I*2	Percent of total hours for each phase
TPHSPC(6)	I*2	Percent of weighted total cost for each phase
TURN	I*2	Flag indicating whether the programmer data are from the CSR or the RSF file = 1, from RSF file = 2, from CSR file
TWTHR(6)	I*2	Weighted total hours by phase
TWTMM(6)	R*4	Weighted total hours in person-months by phase
TWTPCT(6)	I*2	Percent of weighted total hours for each phase
T95T75	R*4	Factor used to convert S/360-95 computer time to S/360-75 time
WEEKPC(6)	I*2	Percent of weeks for each phase
WTLP(3)	I*2	Weighted source lines produced per person-month using programmer time only
WTLPM(3)	I*2	Weighted source lines produced per person-month using programmer and management time
WTLPMS(3)	I*2	Weighted source lines produced per person-month using programmer, management, and services time
XCOST	R*4	Total cost based on weighted hours
XMMM	R*4	Total management hours
XMWTMM	R*4	Total weighted management hours
XPMM	R*4	Total programmer hours
XPWTMM	R*4	Total weighted programmer hours
XSMM	R*4	Total services hours
XSWTMM	R*4	Total weighted services hours

3.3.4 TASK BUILD PROCEDURE

3.3.4.1 Command Procedures

The RU program can be generated from the source code by executing the command procedure RUGEN.CMD under UIC [204,6]. This command procedure references three command files--RUFPP.CMD, RUFOR.CMD, and RU.TKB--all under UIC [204,6]. Figure 3-8 is a listing of RUGEN.CMD, the command procedure to precompile, compile, and task build the RU program. The RU program is generated by entering the following command:

```
@[204,6]RUGEN
```

3.3.4.2 Overlay Structure

The RU program is overlaid to reduce the memory space requirement. Figure 3-9 is a listing of the Overlay Descriptor Language file, [204,6]RU.ODL, needed to build the RU program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay. In addition, the RMS Indexed Access Program Library (RMSIAC) is also needed in the overlay. The name of the library is [204,7]UFRMSIAC.OLB. It contains FORTRAN routines necessary for accessing RMS indexed files.

```
;                                                                          1
;   @RUGEN.CMD                                                             2
;                                                                          3
;   COMMAND PROCDURE TO TASK BUILD THE RESOURCE UTILIZATION (RU) REPORT     4
;   PROGRAM FROM SOURCE  (4/19/82  BY P. LO)                               5
;                                                                          6
;   PRECOMPILE ROUTINES WRITTEN IN STRUCTURED FORTRAN                      7
;                                                                          8
@[204,6]RUFPP.CMD                                                          9
;                                                                         10
;   @RUFPP.CMD                                                            11
;                                                                         12
;   COMMAND PROCEDURE TO PRECOMPILE ROUTINES WRITTEN IN STRUCTURED        13
;   FORTRAN FOR THE RESOURCE UTILIZATION (RU) PROGRAM     4/15/82         14
;                                                                         15
;   ROUTINES WITH PREFIX RU                                               16
;                                                                         17
;FPP SY:[204,6]RUFIGRUN                                                   18
;FPP SY:[204,6]RUFPHASE                                                   19
;FPP SY:[204,6]RUGETCSR                                                   20
;FPP SY:[204,6]RUGETRSF                                                   21
;FPP SY:[204,6]RUHEADER                                                   22
;FPP SY:[204,6]RUMATCH4                                                   23
;FPP SY:[204,6]RUPAGE1                                                    24
;FPP SY:[204,6]RUPRDATA                                                   25
;FPP SY:[204,6]RUPREXTR                                                   26
;FPP SY:[204,6]RUPRPHAS                                                   27
;FPP SY:[204,6]RUPRRUN                                                    28
;FPP SY:[204,6]RUREADNL                                                   29
;FPP SY:[204,6]RUREPORT                                                   30
;FPP SY:[204,6]RURU                                                       31
;FPP SY:[204,6]RUWRTPLT                                                   32
;                                                                         33
;   ROUTINES WITH PREFIX UT                                               34
;                                                                         35
;FPP SY:[204,7]UTDDOPNR                                                   36
;FPP SY:[204,7]UTDOPEN2                                                   37
;FPP SY:[204,7]UTGETFLD                                                   38
;FPP SY:[204,7]UTMAKNAM                                                   39
;FPP SY:[204,7]UTMATCHS                                                   40
;FPP SY:[204,7]UTMOVE                                                     41
;FPP SY:[204,7]UTNAME3                                                    42
;FPP SY:[204,7]UTNEXTWK                                                   43
;FPP SY:[204,7]UTPHSCHK                                                   44
;FPP SY:[204,7]UTRDCSR                                                    45
;FPP SY:[204,7]UTRDEST                                                    46
;FPP SY:[204,7]UTRDHDRX                                                   47
;FPP SY:[204,7]UTRDHDR1                                                   48
;FPP SY:[204,7]UTRDRSF                                                    49
;FPP SY:[204,7]UTSQEEZ                                                    50
;FPP SY:[204,7]UTWHERE                                                    51
;                                                                         52
;   COMPILE FORTRAN SOURCE                                                53
;                                                                         54
@[204,6]RUFOR.CMD                                                         55
```

Figure 3-8. RU Task Generation Command Procedure (RUGEN.CMD) (1 of 2)

```
;
;   @RUFOR.CMD
;
;   COMMAND PROCEDURE TO COMPILE FORTRAN ROUTINES FOR THE RESOURCE
;   UTILIZATION (RU) REPORT PROGRAM       (4/16/82  BY P. LO)
;
;   ROUTINES WITH PREFIX RU
;
;FOR/F4P/OBJECT:[204,6]RUFIGRUN [204,6]RUFIGRUN
;FOR/F4P/OBJECT:[204,6]RUFPHASE [204,6]RUFPHASE
;FOR/F4P/OBJECT:[204,6]RUGETCSR [204,6]RUGETCSR
;FOR/F4P/OBJECT:[204,6]RUGETRSF [204,6]RUGETRSF
;FOR/F4P/OBJECT:[204,6]RUHEADER [204,6]RUHEADER
;FOR/F4P/OBJECT:[204,6]RUMATCH4 [204,6]RUMATCH4
;FOR/F4P/OBJECT:[204,6]RUPAGE1  [204,6]RUPAGE1
;FOR/F4P/OBJECT:[204,6]RUPRDATA [204,6]RUPRDATA
:FOR/F4P/OBJECT:[204,6]RUPREXTR [204,6]RUPREXTR
;FOR/F4P/OBJECT:[204,6]RUPRPHAS [204,6]RUPRPHAS
:FOR/F4P/OBJECT:[204,6]RUPRRUN  [204,6]RUPRRUN
;FOR/F4P/OBJECT:[204,6]RUREADNL [204,6]RUREADNL
;FOR/F4P/OBJECT:[204,6]RUREPORT [204,6]RUREPORT
;FOR/F4P/OBJECT:[204,6]RURU     [204,6]RURU
;FOR/F4P/OBJECT:[204,6]RUWRTPLT [204,6]RUWRTPLT
;
;   ROUTINES WITH PREFIX UT
;
;FOR/F4P/OBJECT:[204,7]UTDDOPNR [204,7]UTDDOPNR
;FOR/F4P/OBJECT:[204,7]UTDOPEN2 [204,7]UTDOPEN2
;FOR/F4P/OBJECT:[204,7]UTGETFLD [204,7]UTGETFLD
;FOR/F4P/OBJECT:[204,7]UTMAKNAM [204,7]UTMAKNAM
;FOR/F4P/OBJECT:[204,7]UTMATCHS [204,7]UTMATCHS
;FOR/F4P/OBJECT:[204,7]UTMOVE   [204,7]UTMOVE
;FOR/F4P/OBJECT:[204,7]UTNAME3  [204,7]UTNAME3
;FOR/F4P/OBJECT:[204,7]UTNEXTWK [204,7]UTNEXTWK
;FOR/F4P/OBJECT:[204,7]UTPHSCHK [204,7]UTPHSCHK
;FOR/F4P/OBJECT:[204,7]UTRDCSR  [204,7]UTRDCSR
;FOR/F4P/OBJECT:[204,7]UTRDEST  [204,7]UTRDEST
;FOR/F4P/OBJECT:[204,7]UTRDHDRX [204,7]UTRDHDRX
;FOR/F4P/OBJECT:[204,7]UTRDHDR1 [204,7]UTRDHDR1
;FOR/F4P/OBJECT:[204,7]UTRDRSF  [204,7]UTRDRSF
;FOR/F4P/OBJECT:[204,7]UTSQEEZ  [204,7]UTSQEEZ
;FOR/F4P/OBJECT:[204,7]UTWHERE  [204,7]UTWHERE
;
;   GENERATE THE TASK IMAGE
;
TKB @[204,6]RU.TKB
;
;   @RU.TKB
;
;   COMMAND PROCEDURE TO TASK BUILD THE RU PROGRAM
;
;[204,5]RU/FU=[204,6]RU/MP
;UNITS=20
;ASG=SY:6
;//
```

Figure 3-8. RU Task Generation Command Procedure (RUGEN.CMD) (2 of 2)

```
;                                                                          1
;   @RU.ODL                                                                2
;                                                                          3
;   RESOURCE UTILIZATION REPORT PROGRAM OVERLAY       4/15/82              4
;                                                                          5
          .ROOT $TREE1,OTSALL,RMSALL                                       6
$TREE1:   .FCTR $ROOT-RMSROT-OTSROT-*($LVL1)                               7
$ROOT:    .FCTR [204,6]RURU-[204,6]RUHEADER-[204,7]UTMOVE-$ROOT1           8
$ROOT1:   .FCTR [204,7]UTRDEST-[204,7]UTSQEEZ-[204,7]UTMATCHS-$ROOT2       9
$ROOT2:   .FCTR [204,7]UTRDHDRX-[204,7]UTRDHDR1-[204,7]UTPHSCHK-$ROOT4    10
$ROOT4:   .FCTR [204,7]UTDDOPNR-[204,7]UTDOPEN2-[204,7]UFRMSIAC/LB        11
;                                                                         12
$LVL1:    .FCTR $RDNL,$RSF,$CSR,[204,6]RUPAGE1,$LVL2                      13
$LVL2:    .FCTR [204,6]RUREPORT-[204,6]RUWRTPLT-[204,7]UTNAME3-($LVL3)    14
$LVL3:    .FCTR [204,6]RUFPHASE,[204,6]RUFIGRUN,[204,6]RUPRDATA-($LVL4)   15
$LVL4:    .FCTR [204,6]RUPRPHAS,[204,6]RUPRRUN,[204,6]RUPREXTR            16
;                                                                         17
$RDNL:    .FCTR [204,6]RUREADNL-[204,7]UTGETFLD-[204,7]UTMAKNAM           18
;                                                                         19
$CSR:     .FCTR [204,6]RUGETCSR-[204,7]UTRDCSR                            20
;                                                                         21
$RSF:     .FCTR [204,6]RUGETRSF-[204,7]UTRDRSF-[204,7]UTNEXTWK-$RSF2      22
$RSF2:    .FCTR [204,6]RUMATCH4                                           23
;                                                                         24
;                                                                         25
@LB:[1,1]RMS11M                                                           26
@LB:[1,1]RMS12X                                                           27
;                                                                         28
          .END                                                            29
```

Figure 3-9. RU Program Overlay Descriptor Language File (RU.ODL)

3.4 WEEKLY HOUR AND FORM COUNT PROGRAM (WK)

3.4.1 INTRODUCTION

The Weekly Hour and Form Count Report Program (WK) produces reports of hour or form counts from a desired SEL data base file for a given project. There are currently 14 different WK reports. Each report contains counts of records, forms, or other data given by programmer by week (Section 2.4).

3.4.2 PROGRAM STRUCTURE

3.4.2.1 Files Accessed

Each of the 14 reports currently produced by the WK program accesses four input files and three output files. All possible files are listed below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file (accessed by all report types)
[204,1]EST.HDR	Estimated Statistics (EST) file (accessed by all report types)
[204,1]HEADER.HDR	Phase Dates (HDR) file (accessed by all report types)
[204,1]<PRJNAM>.ACC	Accounting Information (ACC) file for the given project (accessed by report types XW1, XW2, and XW3)
[204,1]<PRJNAM>.CRF	Change Report Form (CRF) file for the given project (accessed by report type HW)
[204,1]<PRJNAM>.CSF	Component Summary Form (CSF) file for the given project (accessed by report type MW)
[204,1]<PRJNAM>.CSR	Component Status Report (CSR) file for the given project (accessed by report types TH and TW)
[204,1]<PRJNAM>.RAF	Run Analysis Form (RAF) file for the given project (accessed by report types AW1 and AW2)
[204,1]<PRJNAM>.RSF	Resource Summary Form (RSF) file for the given project (accessed by report types RH1, RH2, RH3, RP, and RR)

Output File Name	Description
<PRJNAM>.xxx	Report file for the given project, where xxx = report type (AW1, AW2, HW, MW, RH1, RH2, RH3, RP, RR, TH, TW, XW1, XW2, or XW3)
<PRJNAM>.1xxx	Plot file for the given project for pie chart plotting (not implemented), where xxx = report type
<PRJNAM>.2xxx	Plot file for the given project for graphing (Secton 3.7), where xxx = report type

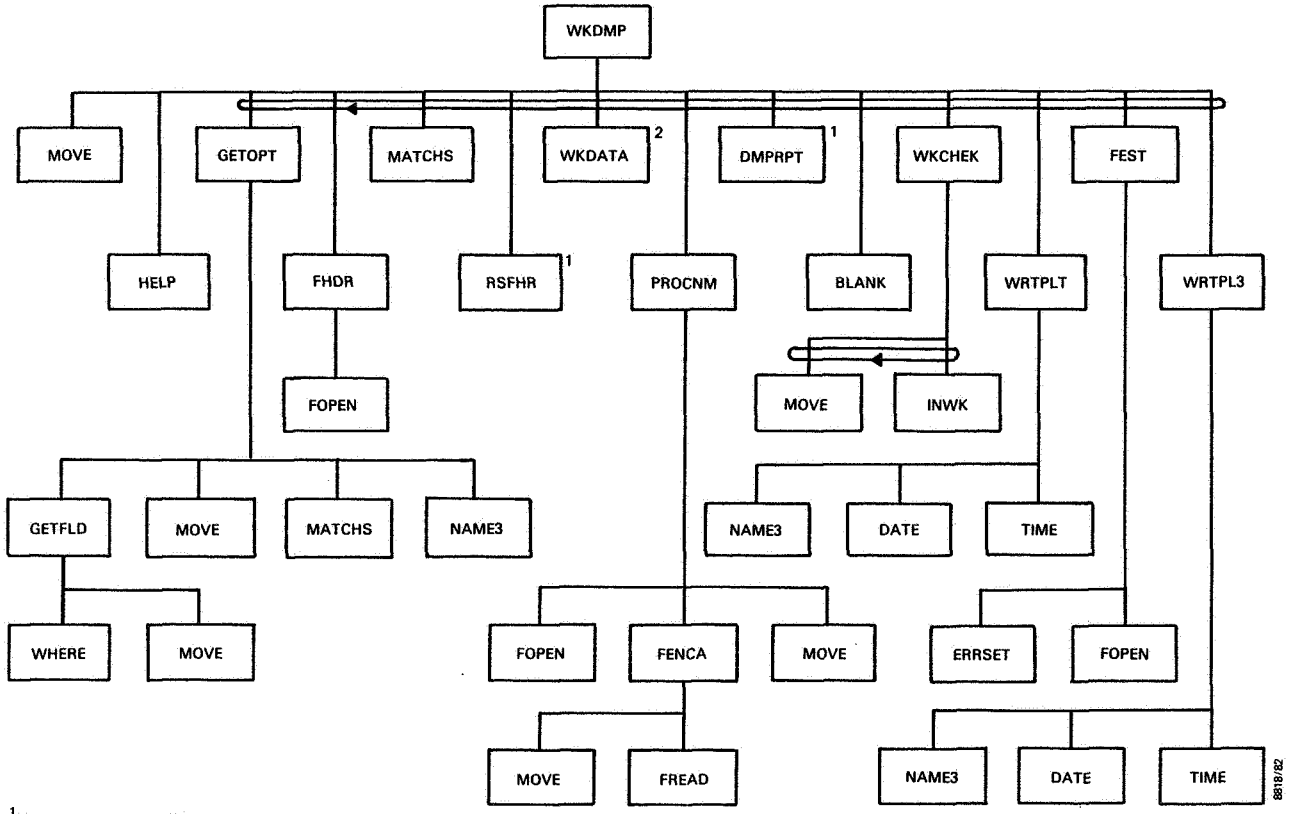
In these files names, <PRJNAM> is the name of the project selected by the user.

3.4.2.2 Baseline Diagram

Figure 3-10 is the baseline diagram for the WK program. The WKDMP routine is the main driver. It displays the help information, obtains the project name and report type, reads the desired file for a given project, reads the HDR and EST files, and produces the report by resource or programmer by week with subtotals given by phase. WKDMP loops through the above process until a ^Z (control Z) is returned by the user in response to a prompt.

3.4.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the WK program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major WK routines are described in Section 3.4.3.7. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the WK program also uses the following system routines: DATE, ERRSET, ERRSNS, and TIME.



[1]SEE FIGURE 3-10 (2 OF 3).

[2]SEE FIGURE 3-10 (3 OF 3).

Figure 3-10. Baseline Diagram for the Weekly Hour and Form Count Report Program (WK) (1 of 3)

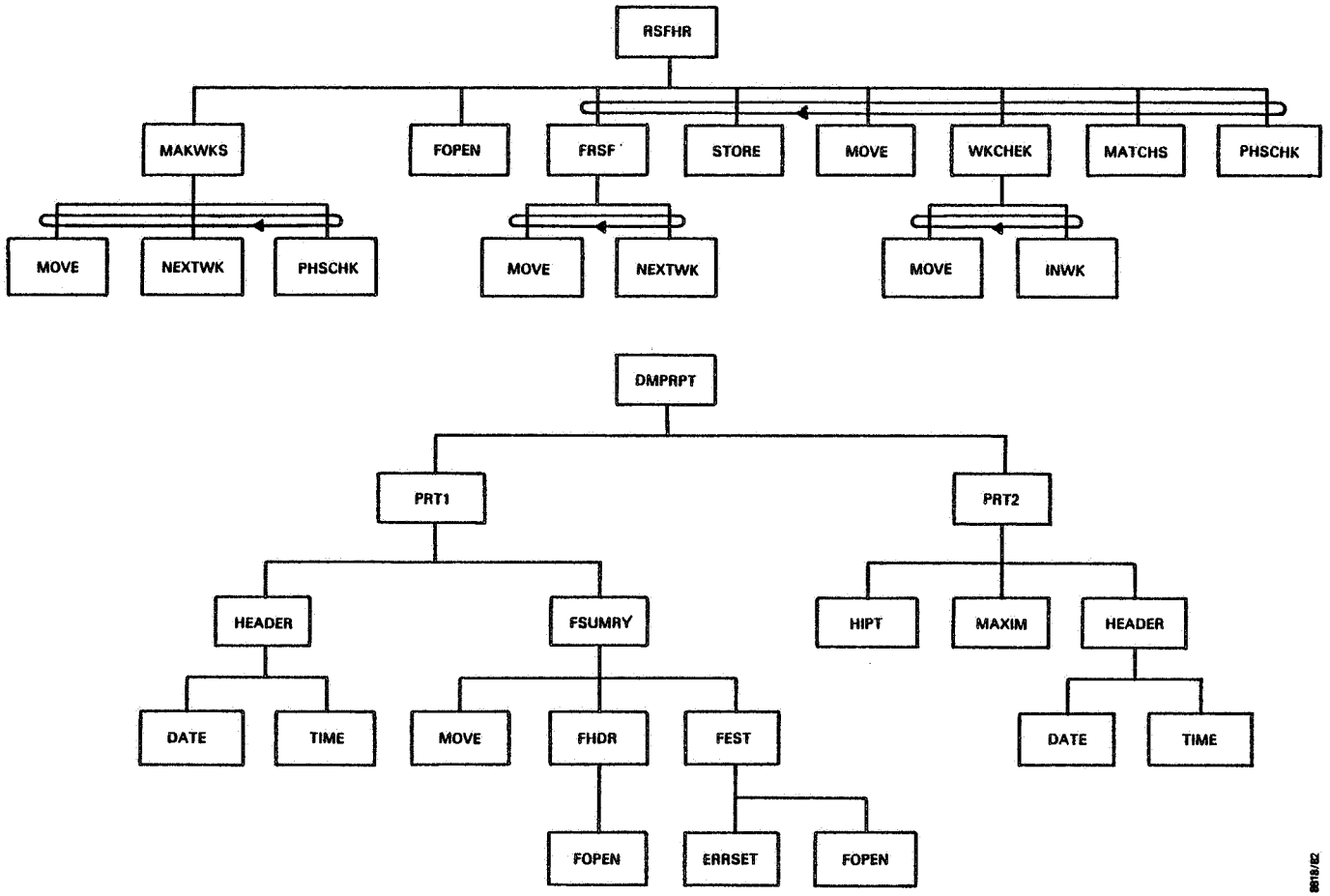


Figure 3-10. Baseline Diagram for the Weekly Hour and Form Count Report Program (WK) (2 of 3)

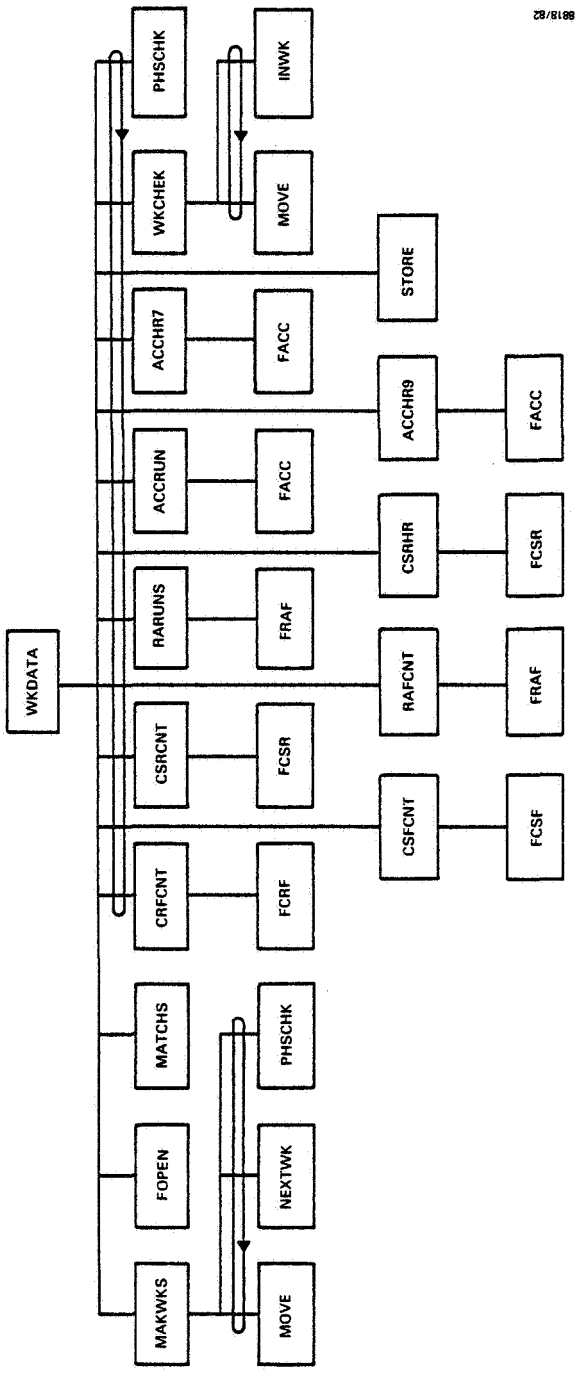


Figure 3-10. Baseline Diagram for the Weekly Hour and Form Count Report Program (WK) (3 of 3)

3.4.3.1 Process Data and Compute Statistics

These twelve major routines obtain data from the given SEL data base file and compute statistics for the WK report.

ROUTINE: ACCHR7

FUNCTION: Reads one ACC file record and returns the date of record, computer code, and IBM S/360-75 time

CALLING SEQUENCE:

```
CALL ACCHR7 (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: ACCHR9

FUNCTION: Reads one ACC file record and returns the date of record, computer code, and IBM S/360-95 time

CALLING SEQUENCE:

```
CALL ACCHR9 (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: ACCRUN

FUNCTION: Reads one record from the ACC file and returns the date of record, computer code, and run count

CALLING SEQUENCE:

```
CALL ACCRUN (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: CRFCNT

FUNCTION: Reads one record from the CRF file and returns the date of form, programmer number, and count

CALLING SEQUENCE:

```
CALL CRFCNT (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: CSFCNT

FUNCTION: Reads one CSF file record and returns the date of form, programmer code, and count

CALLING SEQUENCE:

```
CALL CSFCNT (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: CSRCNT

FUNCTION: Reads one CSR file record and returns the date of form, programmer code, and count

CALLING SEQUENCE:

```
CALL CSRCNT (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: CSRHR

FUNCTION: Reads one CSR file record and returns the date of form, programmer number, and hour count

CALLING SEQUENCE:

```
CALL CSRHR (IDBF,
            DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: MAKWKS

FUNCTION: Sets up an array of weeks covering the given timespan

CALLING SEQUENCE:

```
CALL MAKWKS (DRANG1, DRANG2,
             NWEEKS, WEEKS)
```

ROUTINE: NEXTWK

FUNCTION: Computes date 1 week from the given date in YYMMDD format

CALLING SEQUENCE:

```
CALL NEXTWK (DATE,
             D)
```

ROUTINE: RAFCNT

FUNCTION: Reads one RAF file record and returns the date of form, programmer number, and form count

CALLING SEQUENCE:

```
CALL RAFCNT (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: RARUNS

FUNCTION: Reads one RAF file record and returns the date of form, programmer number, and run count

CALLING SEQUENCE:

```
CALL RARUNS (IDBF,
             DATE, KOUNT, RESID, NULL, EOF, ERROR)
```

ROUTINE: RSFHR

FUNCTION: Accumulates staff hours from the RSF file for each week from the beginning of the design phase to the end of the cleanup phase

CALLING SEQUENCE:

```
CALL RSFHR (DRANG1, DRANG2, IRSFF, KEY, RSFNAM, RSFRUN,
            TYPE,
            AFTTOT, ALLTOT, BEFTOF, HRDATA, NPROG,
            NWEEKS, PHDATA, PHTOT, PRGAFT, PRGBEF,
            PRGTOT, PROGNO, WEEKS, WKTOT, ERROR)
```

ROUTINE: WKDATA

FUNCTION: Accumulates staff hours or counts for each week in the given timespan from the given data base file

CALLING SEQUENCE:

```
CALL WKDATA (DRANG1, DRANG2, IDBF, RSFNAM, TYPE,
             AFTTOT, ALLTOT, BEFTOT, HRDATA, NPROG,
             NWEEKS, PHDATA, PHTOT, PRGAFT, PRGBEF,
             PRGTOT, PROGNO, WEEKS, WKTOT, ERROR)
```

ROUTINE: WKDMP

FUNCTION: Main routine of the WK program, reads the desired file for a given project and produces a report by person by week with subtotals by phase

CALLING SEQUENCE: None

3.4.3.2 Write Output Reports and Plot Files

These seven routines write the output report and plot files.

ROUTINE: DMPRPT

FUNCTION: Prints the complete WK report

CALLING SEQUENCE:

```
CALL DMPRPT (AFTTOT, ALLTOT, BEFTOT, DESCR, DRANG1,
             DRANG2, HRDATA, IRPTF, NPROG, NWEEKS,
             PHDATA, PHTOT, PRGAFT, PRGBEF, PRGTOT,
             PRJNAM, RPTITL, RPTNAM, SRTIDX, WEEKS, WKTOT)
```

ROUTINE: FSUMRY

FUNCTION: Prints a six-line header summary with data from the HDR and EST files

CALLING SEQUENCE:

```
CALL FSUMRY (IRPTF, PRJNAM)
```

OUTINE: HEADER

FUNCTION: Prints a one-line title for each report page, including the date and the project name

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PRJNAM, RPTITL)
```

ROUTINE: PRT1

FUNCTION: Prints the WK report header page

CALLING SEQUENCE:

```
CALL PRT1 (DESCR, DRANG1, DRANG2, IRPTF, NPROG, PRJNAM,
          RPTITL, SRTIDX)
```

ROUTINE: PRT2

FUNCTION: Prints the WK report data page

CALLING SEQUENCE:

```
CALL PRT2 (AFTTOT, ALLTOT, BEFTOT, DESCR, DRANG1,
          DRANG2, HRDATA, IRPTF, NPROG, NWEEKS, PHDATA,
          PHTOT, PRGAFT, PRGBEF, PRGTOT, PRJNAM, RPTITL,
          SRTIDX, WEEKS, WKTOT)
```

ROUTINE: WRTPLT

FUNCTION: Writes the given data to an intermediate file in preparation for pie chart plotting

CALLING SEQUENCE:

```
CALL WRTPLT (DATA, DESCR, EXT, NDATA, PIETTL, PRJNAM,
            RPTITL)
```

ROUTINE: WRTPL3

FUNCTION: Writes the given data to an intermediate file in preparation for graphing

CALLING SEQUENCE:

```
CALL WRTPL3 (DATA, EXT, KLINES, MARKER, NDATA, PIETTL,
             PRJNAM, RPTITL, XMAX, XTITLE, YMAX, YTITLE)
```

ROUTINE: WRTPL3

FUNCTION: Writes the given data to an intermediate file in preparation for graphing

CALLING SEQUENCE:

```
CALL WRTPL3 (DATA, EXT, KLINES, MARKER, NDATA, PIETTL,
             PRJNAM, RPTITL, XMAX, XTITLE, YMAX, YTITLE)
```

3.4.3.3 Obtain Data From Terminal or External File

These five routines obtain information from a user's response to a terminal prompt or from an external file.

ROUTINE: FENCA

FUNCTION: Finds the description field on the Encoding Dictionary corresponding to the given type and code

CALLING SEQUENCE:

```
CALL FENCA (IENCF, TYPE, CODE,
            NAME, REST, FOUND)
```

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR,
             FIELD)
```

ROUTINE: GETOPT

FUNCTION: Obtains the project name from the terminal

CALLING SEQUENCE:

```
CALL GETOPT (TERMNL,
             PRJNAM, RPTITL, RPTNAM, RSFNAM, TYPE, EOF,
             ERROR)
```

ROUTINE: HELP

FUNCTION: Prints help information to the terminal

CALLING SEQUENCE:

```
CALL HELP
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

3.4.3.4 Sort and Search Routines

These seven routines provide some sort and search functions.

ROUTINE: HIPT (INTEGER*2 FUNCTION)

FUNCTION: Finds the first integer having a single significant digit that is greater than the given integer

CALLING SEQUENCE:

```
HIPT(L)
```

ROUTINE: INWK

FUNCTION: Determines whether the given date is within the date range

CALLING SEQUENCE:

```
CALL INWK (DATIN, DATE1, DATE2,
           INWEEK)
```

ROUTINE: MAXIM (INTEGER*2 FUNCTION)

FUNCTION: Finds the maximum number in an array of integers

CALLING SEQUENCE:

```
MAXIM (ARRAY, NARRAY)
```

ROUTINE: PHSCHK

FUNCTION: Determines whether the given date is within the start and end dates of the given range

CALLING SEQUENCE:

```
CALL PHSCHK (FDATE, DRANG1, DRANG2,
             PHNUM, INPHAS)
```

ROUTINE: PROCNM

FUNCTION: Converts given programmer numbers into programmer names

CALLING SEQUENCE:

```
CALL PROCNM (IENCF, NPROG, PROGNO, KTYPE,
             DESCR, SRTIDX, ERROR)
```

ROUTINE: STORE

FUNCTION: Determines whether the given number is in the given array, adds it if it is not, and returns the location of the given number in the given array

CALLING SEQUENCE:

```
CALL STORE (RESID, MAXPRG,
            PROGNO, NPROG,
            IDNUM, BADID)
```

ROUTINE: WKCHEK

FUNCTION: Determines which week in a given array of weeks contains the given date

CALLING SEQUENCE:

```
CALL WKCHEK (DATIN, NWEEKS, WEEKS,
             WKNUM, INWEEK)
```

3.4.3.5 File Open and Read Routines

These ten routines either open an indexed file or read records from an indexed file.

ROUTINE: FACC

FUNCTION: Reads one record from the ACC file

CALLING SEQUENCE:

```
CALL FACC (IACCF,
           PRJCOD, DATE, TIME, TSOFOR, TSOBCK, RJE,
           CRDRDR, CP1, CPU95, IO95, RUNS95, FAIL95, CP2,
           CPU75, IO75, RUNS75, FAIL75, ISTAT, EOF, ERROR)
```

ROUTINE: FCRF

FUNCTION: Reads one record from the CRF file and converts the data to internal format

CALLING SEQUENCE:

```
CALL FCRF (ICRFF,
           FORMNO, PROJNO, PROGNO, FDATE, NCH, NEXAM,
           OVER1, DATDET, DATBEG, EFFORT, CHTYPE, CHCOMP,
           ERRTYP, ERRIN, DATERR, LGCERR, ACTVTY, ISOLTM,
           PATCH, RELOLD, RELNO, RELDAT, CMTREA, CMTDES,
           CMTGEN, STATUS, EOF, ERROR)
```

ROUTINE: FCSF

FUNCTION: Reads one record from the CSF file

CALLING SEQUENCE:

```
    CALL FCSF (ICSFF,
              FORMNO, PROJNO, PROGNO, PROGI, FDATE, FSTAGE,
              COMPCO, PRECIS, CMPLEX, SWTYPE, PASGN, PCNTL,
              POTHER, STATWO, STMT, BTSIZE, INDEP, RELSW,
              ADDTYP, NCALLD, X1, NCALNG, X2, NSHR, X3,
              NDESC, X4, LANG1, PLANG1, LANG2, PLANG2, DES,
              CONSTR, DESRUN, CODRUN, TSTRUN, DESTIM,
              CODTIM, TSTTIM, DESEFF, CODEFF, TSTEFF,
              DESDAT, CODDAT, TSTDAT, DESCR, CALLD, CALNG,
              SHR, AFFECT, OTH, NAMCON, CMT1, CMT2, ISTAT,
              EOF, ERROR)
```

ROUTINE: FCSR

FUNCTION: Reads one record from the CSR file using a FORTRAN read

CALLING SEQUENCE:

```
    CALL FCSR (ICSRF,
              FORMNO, SEQNO, PROJNO, PROGNO, FDATE, COMPCO,
              TIMES, OTHNAM, OTHOUR, ISTAT, PHASE, EOF,
              ERROR)
```

ROUTINE: FEST

FUNCTION: Reads one record from the EST file and converts all data to internal format

CALLING SEQUENCE:

```
    CALL FEST (IESTF, NAME,
              PROJ, NCOMP, MODDEL, MODNEW, MODMOD, NRUNS,
              NCHANG, PAGDOC, LINDEL, LINNEW, LINMOD,
              TOTEXT, NEWEXT, MODEXT, PROGHR, MGMTHR,
              OTHRHR, HR95, HR75, OTHCMP, STATUS, ACTIVE,
              PRJCAT, FOUND, ERROR)
```

ROUTINE: FHDR

FUNCTION: Reads one record from the HDR file and converts all data to internal format

CALLING SEQUENCE:

```
CALL FHDR (IHDRF, PRJNAM,
          PROJ, DEVCMP, TARG, ALIEN, RANGES, STATUS,
          ERROR)
```

ROUTINE: FOPEN

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL FOPEN (IUNIT, FILNAM,
           ERROR)
```

ROUTINE: FRAF

FUNCTION: Reads one record from the RAF file using a FORTRAN read

CALLING SEQUENCE:

```
CALL FRAF (IRAFF,
          FORMNO, SEQNO, PROJNO, PROGNO, RDATE, MACHIN,
          INTERA, PURPOS, NCOMP, COMPCO, FIRST, METOBJ,
          RESULT, COMENT, ISTAT, EOF, ERROR)
```

ROUTINE: FREAD

FUNCTION: Reads one indexed record

CALLING SEQUENCE:

```
CALL FREAD (IUNIT, KEYVAL, KEYLEN, LRECL,
           BUFFER, ERROR)
```

ROUTINE: FRSF

FUNCTION: Reads one record from the RSF file and returns all data on that record plus an array of dates for each week for which there is a resource entry on the record

CALLING SEQUENCE:

```
CALL FRSF (IRSFF,
          FORMNO, SEQNO, PROJNO, RESCOD, RESID, FDATE,
          PCMGMT, WKDATE, NRUNS, TIMES, STATUS, PHASE,
          LASTWK, EOF, ERROR)
```

3.4.3.6 Routines for String Movement or Comparison

These four routines concern string movement or comparison.

ROUTINE: BLANK

FUNCTION: Initializes an array to blanks

CALLING SEQUENCE:

```
CALL BLANK (ARRAY, NUM)
```

ROUTINE: MATCHS (LOGICAL FUNCTION)

FUNCTION: Determines whether the two input strings are the same

CALLING SEQUENCE:

```
MATCHS (ARRAY1, ARRAY2, NBYTES)
```

ROUTINE: MOVE

FUNCTION: Moves given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: WHERE

FUNCTION: Finds the location of the given character in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
            LOC, FOUND)
```

3.4.3.7 Variable Description

The variables in the calling sequences of major WK routines are described below.

Name	Type	Description
AFTTOT	I*2	Total number of programmer hours after cleanup
ALLTOT	I*4	Total programmer hours
ARRAY(NARRAY)	I*2	Array to be searched
BADID	L*1	Error flag to indicate that there is no room for the new number in the given array
BEFTOT	I*2	Total programmer hours before design
DATE(3)	I*2	Form date (YY,MM,DD)
DATE1(3)	I*2	Range start date (YY,MM,DD)
DATE2(3)	I*2	Range end date (YY,MM,DD)
DATIN(3)	I*2	Given date (YY,MM,DD)
DESCR(20,20)	L*1	Programmer names
DRANG1(3,6)	I*2	Phase start dates
DRANG2(3,6)	I*2	Phase end dates
EOF	L*1	End-of-file flag
ERROR	L*1	Error flag
HRDATA(20,400)	I*2	Number of programmer (or other) hours for each week
IDBF	I*2	Unit number for data base file
IDNUM	I*2	Location of given number in array
IENCF	I*2	Unit number for ENC file

Name	Type	Description
INWEEK	L*1	Flag indicating whether given date falls within range
IRPTF	I*2	Unit number for output report file
IRSFF	I*2	Unit number for RSF file
KEY	L*1	Code used to determine which resource is desired M = manpower C = computer O = other (services)
KOUNT	I*2	Hour, person, or run count for given record
KTYPE	I*2	Resource type
L	I*2	Given number
MAXPRG	I*2	Maximum number of array elements allowed
NARRAY	I*2	Size of array
NPROG	I*2	Number of programmers
NULL	L*1	Flag indicating whether record read is usable
NWEEKS	I*2	Number of weeks in project
PHDATA(20,5)	I*2	Phase subtotals
PHTOT(5)	I*2	Phase totals
PRGAFT(20)	I*2	Programmer totals after cleanup
PRGBEF(20)	I*2	Programmer totals before design
PRGTOT(20)	I*2	Totals for each programmer
PRJNAM(8)	L*1	Project name
PROGNO(20)	I*4	Programmer numbers
RESID	I*4	Programmer or computer code
RPTITL(40)	L*1	Report title
RPTNAM(27)	L*1	Report file name
RSFNAM(27)	L*1	Data base file name
RSFRUN	L*1	Flag indicating that RSF file run count is desired
SRTIDX(20)	I*2	Sorted index array to alphabetize programmers
TERMNL	L*1	Flag of whether to read from terminal or external file

Name	Type	Description
TYPE(3)	L*1	Report type
WEEKS(3,400)	I*2	Week array
WKNUM	I*2	Number of week containing given date = 0 if given date is after range = -1 if given date is before range
WKTOT(400)	I*2	Total hours each week

3.4.4 TASK BUILD PROCEDURE

3.4.4.1 Command Procedures

The WK program can be generated from the source code by executing the command procedure WKGEN.CMD under UIC [204,6]. This command procedure references three command files--WKFPP.CMD, WKFOR.CMD, and WK.TKB--all under UIC [204,6]. Figure 3-11 is a listing of the command procedure WKGEN.CMD, which precompiles, compiles, and builds the WK program task image. The WK program is generated by entering the following command:

```
@[204,6]WKGEN
```

3.4.4.2 Overlay Structure

The WK program is overlaid to reduce the memory space requirement. Figure 3-12 is a listing of the Overlay Descriptor Language file, [204,6]WK.ODL, needed to build the WK program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @WKGEN.CMD                                                             2
;                                                                          3
;   GENERATE THE WEEKLY FORM AND HOUR COUNT REPORT PROGRAM (WK)            4
;   TASK   (P. LO    5/10/82)                                              5
;                                                                          6
;   PRECOMPILE FORTRAN SOURCE                                              7
;                                                                          8
@[204,6]WKFPP.CMD                                                          9
;                                                                         10
;   @WKFPP.CMD                                                            11
;                                                                         12
;   PRECOMPILE FORTRAN SOURCE FOR THE WEEKLY FORM AND HOUR COUNT          13
;   REPORT PROGRAM (WK)    (P. LO    5/10/82)                             14
;                                                                         15
;   ROUTINE WITH PREFIX WK                                                16
;                                                                         17
;FPP SY:[204,6]WKACCHR7                                                   18
;FPP SY:[204,6]WKACCHR9                                                   19
;FPP SY:[204,6]WKACCRUN                                                   20
;FPP SY:[204,6]WKCRFCNT                                                   21
;FPP SY:[204,6]WKCSFCNT                                                   22
;FPP SY:[204,6]WKCSRCNT                                                   23
;FPP SY:[204,6]WKCSRHR                                                    24
;FPP SY:[204,6]WKDMPRPT                                                   25
;FPP SY:[204,6]WKGETOPT                                                   26
;FPP SY:[204,6]WKHELP                                                     27
;FPP SY:[204,6]WKHIPT                                                     28
;FPP SY:[204,6]WKINWK                                                     29
;FPP SY:[204,6]WKMAKWKS                                                   30
;FPP SY:[204,6]WKMAXIM                                                    31
;FPP SY:[204,6]WKPROCNM                                                   32
;FPP SY:[204,6]WKPRT1                                                     33
;FPP SY:[204,6]WKPRT2                                                     34
;FPP SY:[204,6]WKRAFCNT                                                   35
;FPP SY:[204,6]WKRARUNS                                                   36
;FPP SY:[204,6]WKRSFHR                                                    37
;FPP SY:[204,6]WKSTORE                                                    38
;FPP SY:[204,6]WKWKCHEK                                                   39
;FPP SY:[204,6]WKWKDATA                                                   40
;FPP SY:[204,6]WKWKDMP                                                    41
;                                                                         42
;   ROUTINE WITH PREFIX UT                                                43
;                                                                         44
;FPP SY:[204,7]UTBLANK                                                    45
;FPP SY:[204,7]UTFACC                                                     46
;FPP SY:[204,7]UTFCRF                                                     47
;FPP SY:[204,7]UTFCSF                                                     48
;FPP SY:[204,7]UTFCSR                                                     49
;FPP SY:[204,7]UTFENCA                                                    50
;FPP SY:[204,7]UTFEST                                                     51
;FPP SY:[204,7]UTFHDR                                                     52
;FPP SY:[204,7]UTFOPEN                                                    53
;FPP SY:[204,7]UTFRAF                                                     54
;FPP SY:[204,7]UTFREAD                                                    55
```

Figure 3-11. WK Task Generation Command Procedure (WKGEN.CMD) (1 of 3)

```
;FPP SY:[204,7]UTFRSF                                                    56
;FPP SY:[204,7]UTFSUMRY                                                  57
;FPP SY:[204,7]UTGETFLD                                                  58
;FPP SY:[204,7]UTHEADER                                                  59
;FPP SY:[204,7]UTMATCHS                                                  60
;FPP SY:[204,7]UTMOVE                                                    61
;FPP SY:[204,7]UTNAME3                                                   62
;FPP SY:[204,7]UTNEXTWK                                                  63
;FPP SY:[204,7]UTPHSCHK                                                  64
;FPP SY:[204,7]UTSQEEZ                                                   65
;FPP SY:[204,7]UTWHERE                                                   66
;FPP SY:[204,7]UTWRTPLT                                                  67
;FPP SY:[204,7]UTWRTPL3                                                  68
;                                                                        69
;   COMPILE FORTRAN SOURCE                                               70
;                                                                        71
@[204,6]WKFOR.CMD                                                        72
;                                                                        73
;   @WKFOR.CMD                                                           74
;                                                                        75
;   COMPILE FORTRAN SOURCE FOR THE WEEKLY FORM AND HOUR COUNT            76
;   REPORT PROGRAM (WK)   (P. LO    5/10/82)                             77
;                                                                        78
;   ROUTINE WITH PREFIX WK                                               79
;                                                                        80
;FOR/F4P/OBJECT:[204,6]WKACCHR7 [204,6]WKACCHR7                           81
;FOR/F4P/OBJECT:[204,6]WKACCHR9 [204,6]WKACCHR9                           82
;FOR/F4P/OBJECT:[204,6]WKACCRUN [204,6]WKACCRUN                           83
;FOR/F4P/OBJECT:[204,6]WKCRFCNT [204,6]WKCRFCNT                           84
;FOR/F4P/OBJECT:[204,6]WKCSFCNT [204,6]WKCSFCNT                           85
;FOR/F4P/OBJECT:[204,6]WKCSRCNT [204,6]WKCSRCNT                           86
;FOR/F4P/OBJECT:[204,6]WKCSRHR  [204,6]WKCSRHR                            87
;FOR/F4P/OBJECT:[204,6]WKDMPRPT [204,6]WKDMPRPT                           88
;FOR/F4P/OBJECT:[204,6]WKGETOPT [204,6]WKGETOPT                           89
;FOR/F4P/OBJECT:[204,6]WKHELP   [204,6]WKHELP                             90
;FOR/F4P/OBJECT:[204,6]WKHIPT   [204,6]WKHIPT                             91
;FOR/F4P/OBJECT:[204,6]WKINWK   [204,6]WKINWK                             92
;FOR/F4P/OBJECT:[204,6]WKMAKWKS [204,6]WKMAKWKS                           93
;FOR/F4P/OBJECT:[204,6]WKMAXIM  [204,6]WKMAXIM                            94
;FOR/F4P/OBJECT:[204,6]WKPROCNM [204,6]WKPROCNM                           95
;FOR/F4P/OBJECT:[204,6]WKPRT1   [204,6]WKPRT1                             96
;FOR/F4P/OBJECT:[204,6]WKPRT2   [204,6]WKPRT2                             97
;FOR/F4P/OBJECT:[204,6]WKRAFCNT [204,6]WKRAFCNT                           98
;FOR/F4P/OBJECT:[204,6]WKRARUNS [204,6]WKRARUNS                           99
;FOR/F4P/OBJECT:[204,6]WKRSFHR  [204,6]WKRSFHR                           100
;FOR/F4P/OBJECT:[204,6]WKSTORE  [204,6]WKSTORE                           101
;FOR/F4P/OBJECT:[204,6]WKWKCHEK [204,6]WKWKCHEK                          102
;FOR/F4P/OBJECT:[204,6]WKWKDATA [204,6]WKWKDATA                          103
;FOR/F4P/OBJECT:[204,6]WKWKDMP  [204,6]WKWKDMP                           104
;                                                                       105
;   ROUTINE WITH PREFIX UT                                              106
;                                                                       107
;FOR/F4P/OBJECT:[204,7]UTBLANK  [204,7]UTBLANK                          108
;FOR/F4P/OBJECT:[204,7]UTFACC   [204,7]UTFACC                           109
;FOR/F4P/OBJECT:[204,7]UTFCRF   [204,7]UTFCRF                           110
```

Figure 3-11. WK Task Generation Command Procedure (WKGEN.CMD) (2 of 3)

```
;FOR/F4P/OBJECT:[204,7]UTFCSF    [204,7]UTFCSF
;FOR/F4P/OBJECT:[204,7]UTFCSR    [204,7]UTFCSR
;FOR/F4P/OBJECT:[204,7]UTFENCA   [204,7]UTFENCA
;FOR/F4P/OBJECT:[204,7]UTFEST    [204,7]UTFEST
;FOR/F4P/OBJECT:[204,7]UTFHDR    [204,7]UTFHDR
;FOR/F4P/OBJECT:[204,7]UTFOPEN   [204,7]UTFOPEN
;FOR/F4P/OBJECT:[204,7]UTFRAF    [204,7]UTFRAF
;FOR/F4P/OBJECT:[204,7]UTFREAD   [204,7]UTFREAD
;FOR/F4P/OBJECT:[204,7]UTFRSF    [204,7]UTFRSF
;FOR/F4P/OBJECT:[204,7]UTFSUMRY  [204,7]UTFSUMRY
;FOR/F4P/OBJECT:[204,7]UTGETFLD  [204,7]UTGETFLD
;FOR/F4P/OBJECT:[204,7]UTHEADER  [204,7]UTHEADER
;FOR/F4P/OBJECT:[204,7]UTMATCHS  [204,7]UTMATCHS
;FOR/F4P/OBJECT:[204,7]UTMOVE    [204,7]UTMOVE
;FOR/F4P/OBJECT:[204,7]UTNAME3   [204,7]UTNAME3
;FOR/F4P/OBJECT:[204,7]UTNEXTWK  [204,7]UTNEXTWK
;FOR/F4P/OBJECT:[204,7]UTPHSCHK  [204,7]UTPHSCHK
;FOR/F4P/OBJECT:[204,7]UTSQEEZ   [204,7]UTSQEEZ
;FOR/F4P/OBJECT:[204,7]UTWHERE   [204,7]UTWHERE
;FOR/F4P/OBJECT:[204,7]UTWRTPLT  [204,7]UTWRTPLT
;FOR/F4P/OBJECT:[204,7]UTWRTPL3  [204,7]UTWRTPL3
;
;   GENERATE THE WK TASK
;
TKB @[204,6]WK.TKB
;
;   @WK.TKB
;
;   TASK BUILD COMMAND PROCEDURE FOR THE WEEKLY FORM AND HOUR COUNT
;   REPORT PROGRAM (WK)
;
:[204,5]WK=[204,6]WK/MP
:UNITS=20
:MAXBUF=250
://
```

Figure 3-11. WK Task Generation Command Procedure (WKGEN.CMD) (3 of 3)

```
;                                                                              1
;   @WK.ODL                                                                    2
;                                                                              3
;   OVERLAY STRUCTURE FOR THE WEEKLY HOUR AND FORM COUNT REPORT PROGRAM        4
;   (WK)                                                                       5
;                                                                              6
        .ROOT $TREE1,RMSALL,OTSALL                                             7
        .NAME FD                                                               8
$TREE1: .FCTR       $ROOT-RMSROT-OTSROT-$ROT20                                 9
$ROOT:  .FCTR       [204,6]WKWKDMP -[204,6]WKWKCHEK-$ROOT6                    10
$ROOT6: .FCTR       [204,7]UTBLANK -$ROOT7                                    11
$ROOT7: .FCTR       [204,7]UTNAME3 -$ROOT8                                    12
$ROOT8: .FCTR       [204,7]UTMATCHS-[204,7]UTMOVE  -$ROT12                    13
$ROT12: .FCTR       [204,7]UTFOPEN -[204,7]UTFREAD -$ROT14                    14
$ROT14: .FCTR       [204,7]UTSQEEZ -[204,7]UTGETFLD-[204,7]UTWHERE            15
$ROT20: .FCTR       *($HLP,$OPT,$FILE,$PROC,$DMP,$PLT)                        16
;                                                                             17
$HLP:     .FCTR     [204,6]WKHELP                                             18
;                                                                             19
$OPT:     .FCTR     [204,6]WKGETOPT                                           20
;                                                                             21
$FILE:    .FCTR     FD-[204,6]WKINWK-[204,6]WKMAKWKS-$FD2                     22
$FD2:     .FCTR     [204,6]WKSTORE-[204,7]UTPHSCHK-$FD3                       23
$FD3:     .FCTR     [204,7]UTNEXTWK-($RSF,$DATA)                              24
;                                                                             25
$RSF:       .FCTR   [204,6]WKRSFHR -[204,7]UTFRSF                             26
;                                                                             27
$DATA:      .FCTR   [204,6]WKWKDATA-($HW,$MW,$TW,$RW,$RW2,$TH,$XW1,$XW2,      28
$HW:          .FCTR [204,6]WKCRFCNT-[204,7]UTFCRF                             29
$MW:          .FCTR [204,6]WKCSFCNT-[204,7]UTFCSF                             30
$TW:          .FCTR [204,6]WKCSRCNT-[204,7]UTFCSR                             31
$RW:          .FCTR [204,6]WKRAFCNT-[204,7]UTFRAF                             32
$RW2:         .FCTR [204,6]WKRARUNS-[204,7]UTFRAF                             33
$TH:          .FCTR [204,6]WKCSRHR -[204,7]UTFCSR                             34
$XW1:         .FCTR [204,6]WKACCRUN-[204,7]UTFACC                             35
$XW2:         .FCTR [204,6]WKACCHR7-[204,7]UTFACC                             36
$XW3:         .FCTR [204,6]WKACCHR9-[204,7]UTFACC                             37
;                                                                             38
$PROC:    .FCTR     [204,6]WKPROCNM-[204,7]UTFENCA                            39
;                                                                             40
$DMP:     .FCTR     [204,6]WKDMPRPT-$A-([204,6]WKPRT1-$OUT, $PRT2)            41
$A:       .FCTR     [204,7]UTHEADER                                           42
$OUT:     .FCTR     [204,7]UTFSUMRY-[204,7]UTFEST-[204,7]UTFHDR               43
$PRT2:    .FCTR     [204,6]WKPRT2  -[204,6]WKMAXIM -[204,6]WKHIPT             44
;                                                                             45
$PLT:     .FCTR     [204,7]UTWRTPLT-[204, 7]UTWRTPL3                          46
;                                                                             47
@LB:[1,1]RMS11M.ODL                                                           48
@LB:[1,1]RMS12X.ODL                                                           49
        .END                                                                  50
```

Figure 3-12. WK Program Overlay Descriptor Language File (WK.ODL)

3.5 COMPONENT INFORMATION REPORT BY FUNCTION TYPE PROGRAM (REP4) AND ITS PREPROCESSOR, THE CHANGE AND ERROR ACCUMULATION PROGRAM (CG)

3.5.1 INTRODUCTION

The Component Information Report by Function Type Program (REP4) produces a list of components and associated data for a given project, organized by the function type of the component and sorted by the number of executable statements. The change and error data on this report are read from an intermediate file produced by the Change and Error Accumulation Program (CG).

3.5.2 PROGRAM STRUCTURE

3.5.2.1 Files Accessed

The CG program accesses two input files and two output files, as described below.

Input File Name	Description
[204,1]<PRJNAM>.CIF	Component Information File (CIF) for the given project
[204,1]<PRJNAM>.CRF	Change Report Form (CRF) file for the given project

Output File Name	Description
<PRJNAM>.CHN	CG intermediate output file containing change and error data for the given project
FOR006.DAT	File containing all component names not found on the CIF for the given project

The REP4 program accesses two input files and one output file, as described below.

Input File Name	Description
[204,1]<PRJNAM>.CIF	CIF for the given project
<PRJNAM>.CHN	CG intermediate file containing change and error data for the given project

Output File Name	Description
<PRJNAM>.RP4	Output report for the given project

In these file names, <PRJNAM> denotes the name of the project selected by the user.

3.5.2.2 Baseline Diagrams

Figure 3-13 is the baseline diagram for the CG program. The XCH routine is the main driver. It obtains the project name, reads the CIF and the CRF file for the given project, accumulates the change and error data from the CRF file, and writes the output files. XCH loops through this process until a^Z (control Z) is returned in response to the prompt for the project name.

Figure 3-14 is the baseline diagram for the REP4 program. The driver routine, REP4, obtains the project name and selected subsystem, reads the CG intermediate file and the CIF for the given project, determines the component type, sorts all components by number of executable statements, and writes the output report. REP4 loops through this process until a^Z (control Z) is returned in response to a subsystem prefix prompt.

3.5.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines referenced by the CG and REP4 programs are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major CG and REP4 routines are described in Section 3.5.3.5. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the CG and REP4 programs also reference the following system routines: DATE, ERRSNS, SECNDS, and TIME.

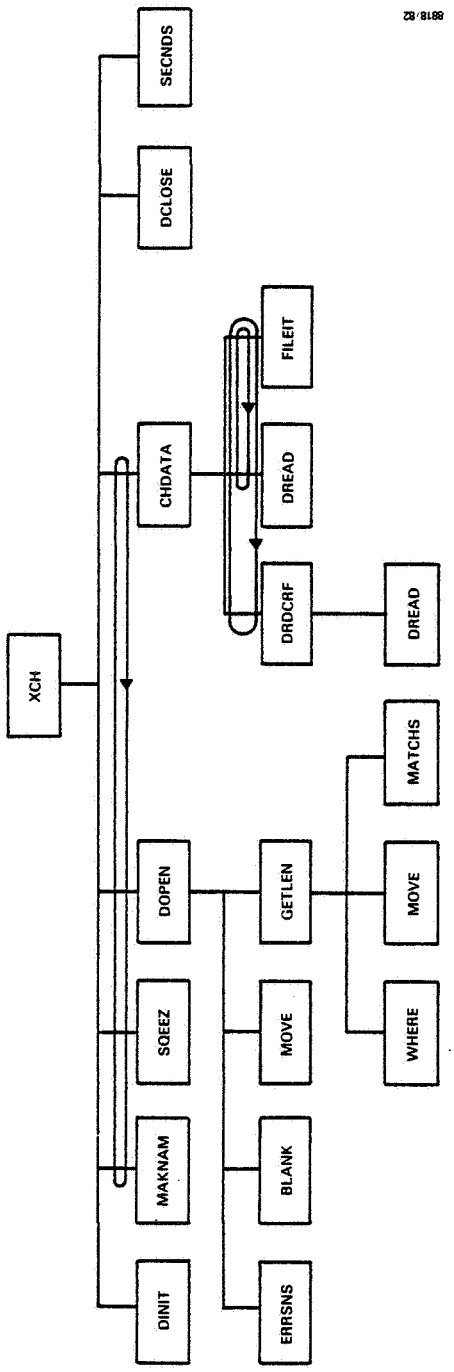


Figure 3-13. Baseline Diagram for the Change and Error Accumulation Program (CG)

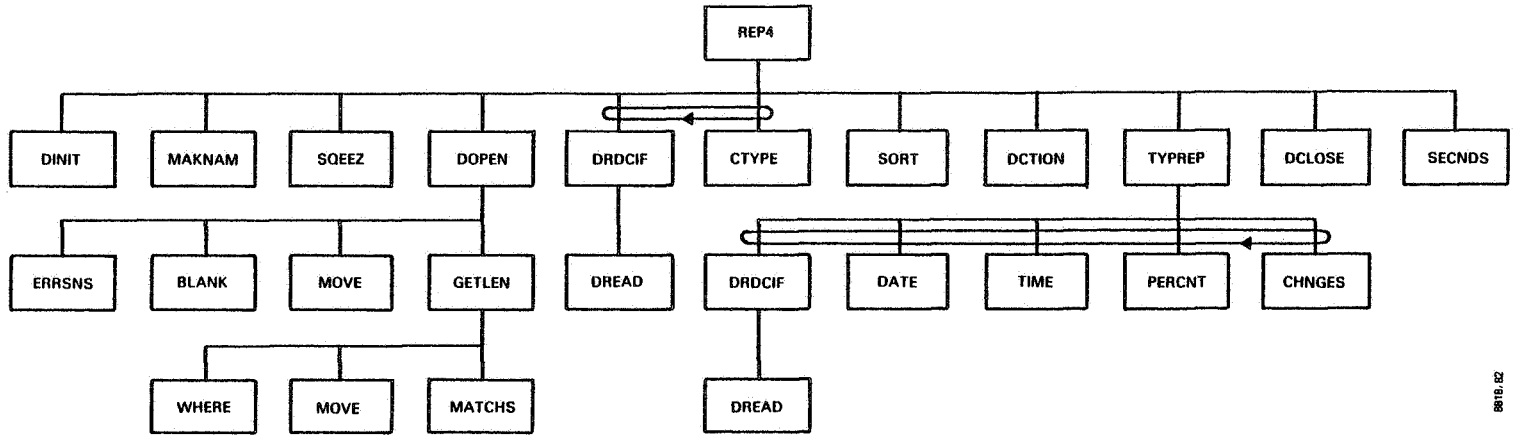


Figure 3-14. Baseline Diagram for the Component Information Report by Function Type Program (REP4)

3.5.3.1 Process Data and Compute Statistics

These six major routines obtain data from a given CIF or CRF file and compute statistics for the CG or the REP4 program.

ROUTINE: CHDATA

FUNCTION: Accumulates change and error data by component from the CRF file

CALLING SEQUENCE:

```
CALL CHDATA (LUNIT, MUNIT, NUNIT, OUTDSN)
```

ROUTINE: CTYPE

FUNCTION: Determines the function type of a component

CALLING SEQUENCE:

```
CALL CTYPE (ICTEXC, ICTFNR, ICTIO, KASGN, KCALL, KFMT,
            ITYPE)
```

ROUTINE: PERCNT

FUNCTION: Computes percentages of several statistics

CALLING SEQUENCE:

```
CALL PERCNT (ICTDOS, ICTEXC, ICTFNR, ICTIFF, ICTIO,
             IDECIS, KASGN, KCALL, KFMT,
             PASGN, PCALL, PDEC, PDOS, PFUNC, PIFS, PIO,
             PTOTS)
```

ROUTINE: REP4

FUNCTION: Main routine of the REP4 program, extracts data from the CIF and the CG intermediate file, determines the function type of the components, and writes the output report

CALLING SEQUENCE: None

ROUTINE: TYPREP

FUNCTION: Reads records from the CIF, computes statistics, and writes the report subdivided by function type of component

CALLING SEQUENCE:

```
CALL TYPREP (ICHNGF, IREPF, LUNDB, ISORT, ITYPE, NSORT,
             ZPROJ, PREFIX, INAME)
```

ROUTINE: XCH

FUNCTION: Main routine of the CG program, accumulates change and error data from the CRF file and writes it to an intermediate output file

CALLING SEQUENCE: None

3.5.3.2 Input and Output Routines

These four routines perform input or output functions.

ROUTINE: CHNGES

FUNCTION: Reads the CG intermediate data file and returns the number of changes and errors for a given component name; if the component name is not found, the routine returns 999 for the output variables

CALLING SEQUENCE:

```
CALL CHNGES (ANAME, ICHNGF,
             NCHS, NERRS, TOTCH)
```

ROUTINE: DCTION

FUNCTION: Prints dictionary of abbreviations for page 1 of the REP4 report

CALLING SEQUENCE:

```
CALL DCTION (PROJ)
```

ROUTINE: DRDCIF

FUNCTION: Reads one record from the CIF and converts all data to internal format

CALLING SEQUENCE:

```
CALL DRDCIF (LUNIT, IKEY, KEYVAL,
             PROJNO, CPREFX, CNAME, ICODE, PANV, MODFUN,
             SYSFUN, ORIGIN, NEXEC, NLINES, NCOMNT,
             IETA1, IETA2, NETA1, NETA2, NIOVAR, NDECIS,
             NFUNCT, NIO, NASGN, NCALL, NFMT, EOF, ERROR,
             LEN)
```

ROUTINE: DRDCRF

FUNCTION: Reads one record from the CRF file and converts all data to internal format

CALLING SEQUENCE:

```
CALL DRDCRF (MUNIT,
             FORMNO, PROJNO, PROGNO, FDATE, NCH, NEXAM,
             OVER1, DATDET, DATBEG, EFFORT, CHTYPE,
             CHCOMP, ERRTYP, ERRIN, DATERR, LGCERR,
             ACTVTY, ISOLTM, PATCH, RELOLD, RELNO,
             RELDAT, CMTREA, CMTDES, CMTGEN, ISTAT, EOF,
             ERROR)
```

3.5.3.3 Sort and Search Routines

These two routines perform sort or search functions.

ROUTINE: FILEIT

FUNCTION: Determines if the given name is in the current list and adds it if it is not

CALLING SEQUENCE:

```
CALL FILEIT (ERRIN, MAXNAM, NAME,
             NEWCH, NEWERR, NEWNAM, NNEW,
             ERROR)
```

ROUTINE: SORT

FUNCTION: Produces an array of indexes sorted in order based on the given I*2 array

CALLING SEQUENCE:

```
CALL SORT (I2, NSORT,
           ISORT)
```

3.5.3.4 Routines Performing String Operations

These two routines perform string operations.

ROUTINE: MAKNAM

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL MAKNAM (DISK, UIC, NAME, EXTENS,
             DSN)
```

ROUTINE: SQEEZ

FUNCTION: Removes blanks from a character string

CALLING SEQUENCE:

```
CALL SQEEZ (IN, NSIZE,
            NONBL, OUT)
```

3.5.3.5 Variable Description

The variables in the calling sequences of major CG and REP4 routines are described below.

Name	Type	Description
ANAME	R*8	Component name
ERRIN	I*2	Flag indicating when error entered system
ERROR	L*1	Error flag
ICHNGF	I*2	Change and error data file (CG intermediate file) unit number

Name	Type	Description
ICTDOS	I*2	Number of DO and DOWHILE statements
ICTEXC	I*2	Number of executable statements
ICTFNR	I*2	Number of FUNCTION references
ICTIFF	I*2	Number of IF and .IF statements
ICTIO	I*2	Number of I/O statements
IDECIS	I*2	Number of decisions (McCabe's measure)
INAME(NSORT)	R*8	Array of names of each component
IREPF	I*2	REP4 output report file unit number
ISORT(NSORT)	I*2	Sorted index array
ITYPE(NSORT)	I*2	Function type of each component
I2(NSORT)	I*2	Array on which sort is based
KASGN	I*2	Number of assignment statements
KCALL	I*2	Number of CALLs
KFMT	I*2	Number of FORMAT statements
LUNDB	I*2	CIF unit number
LUNIT	I*2	Unit number associated with the CIF
MAXNAM	I*2	Maximum number of component names allowed in name array
MUNIT	I*2	Unit number associated with the CRF file
NAME	R*8	Component name
NCHS	I*2	Number of changes for the given component
NERRS	I*2	Number of errors for the given component
NEWCH(NNEW)	I*2	Array of number of changes for each component
NEWERR(NNEW)	I*2	Array of number of errors for each component
NEWNAM(NNEW)	R*8	Array of component names identified in the CRF file
NNEW	I*2	Number of components identified in the CRF file
NSORT	I*2	Number of records to be sorted

Name	Type	Description
NUNIT	I*2	Unit number of the CG intermediate file
OUTDSN(25)	L*1	File name of the CG intermediate file
PASGN	R*4	Percent of assignment statements
PCALL	R*4	Percent of CALLs
PDEC	R*4	Percent of decisions
PDOS	R*4	Percent of DO and DOWHILE statements
PFUNC	R*4	Percent of FUNCTION references
PIFS	R*4	Percent of IF and .IF statements
PIO	R*4	Percent of I/O plus FORMAT statements
PREFIX	I*2	Subsystem prefix given by user
PROJ(8)	L*1	Project name
PTOTS	R*4	Percent of CALL statements plus FUNCTION references
TOTCH	I*2	Total number of changes and errors
ZPROJ(NSORT)	I*2	Subsystem prefix for each component

3.5.4 TASK BUILD PROCEDURE

3.5.4.1 Command Procedures

The CG program can be generated from the source code by executing the command procedure CGGEN.CMD under UIC [204,6] (Figure 3-15). CGGEN.CMD references another command procedure, CG.TKB, under UIC [204,6], which builds the task image for the CG program.

The REP4 program can be generated from the source code by executing the command procedure R4GEN.CMD under UIC [204,6] (Figure 3-16). Three other command procedures, R4FPP.CMD, R4FOR.CMD, and R4.TKB, under UIC [204,6], are referenced by this command procedure.

```
;                                                                          1
;   @CGGEN.CMD                                                             2
;                                                                          3
;   COMMAND PROCEDURE TO PRECOMPILE, COMPILE, AND TASK BUILD THE CHANGE    4
;   AND ERROR ACCUMULATION PROGRAM (CG)   (P. LO    5/26/82)               5
;                                                                          6
;   PRECOMPILE FORTRAN ROUTINES                                            7
;                                                                          8
FPP SY:[204,6]CGCHDATA                                                     9
FPP SY:[204,6]CGFILEIT                                                    10
FPP SY:[204,6]CGXCH                                                       11
FPP SY:[204,7]UTDRDCRF                                                    12
FPP SY:[204,7]UTMAKNAM                                                    13
FPP SY:[204,7]UTSQEEZ                                                     14
;                                                                         15
;   COMPILE FORTRAN ROUTINES                                              16
;                                                                         17
FOR/F4P/OBJECT:[204,6]CGCHDATA [204,6]CGCHDATA                            18
FOR/F4P/OBJECT:[204,6]CGFILEIT [204,6]CGFILEIT                            19
FOR/F4P/OBJECT:[204,6]CGXCH    [204,6]CGXCH                               20
FOR/F4P/OBJECT:[204,7]UTDRDCRF [204,7]UTDRDCRF                            21
FOR/F4P/OBJECT:[204,7]UTMAKNAM [204,7]UTMAKNAM                            22
FOR/F4P/OBJECT:[204,7]UTSQEEZ  [204,7]UTSQEEZ                             23
;                                                                         24
;   BUILD THE CG PROGRAM TASK IMAGE                                       25
;                                                                         26
TKB @[204,6]CG.TKB                                                        27
;                                                                         28
;   @CG.TKB                                                               29
;                                                                         30
;   TASK BUILD THE CHANGE AND ERROR ACCUMULATION PROGRAM (CG)             31
;   (P. LO    5/20/82)                                                    32
;                                                                         33
:[204,5]CG/FU=[204,6]CG/MP                                                34
:ACTFIL=4                                                                 35
:UNITS=20                                                                 36
:ASG=SY:1:2:6:13,TI:5                                                     37
;//                                                                       38
```

Figure 3-15. CG Task Generation Command Procedure (CGGEN.CMD)

```
;                                                                    1
;   @R4GEN.CMD                                                       2
;                                                                    3
;   COMMAND PROCEDURE TO PRECOMPILE, COMPILE, AND TASK BUILD THE     4
;   COMPONENT INFORMATION REPORT BY TYPE PROGRAM (REP4)               5
;   (P. LO    5/26/82)                                               6
;                                                                    7
;   PRECOMPILE FORTRAN ROUTINES                                      8
;                                                                    9
@[204,6]R4FPP.CMD                                                   10
;                                                                   11
;   @R4FPP.CMD                                                      12
;                                                                   13
;   PRECOMPILE FORTRAN ROUTINES FOR THE COMPONENT INFORMATION REPORT BY  14
;   TYPE PROGRAM (REP4)   (P. LO    5/26/82)                        15
;                                                                   16
;   ROUTINES WITH PREFIX R4                                         17
;                                                                   18
;FPP SY:[204,6]R4DCTION                                             19
;FPP SY:[204,6]R4PERCNT                                             20
;FPP SY:[204,6]R4REP4                                               21
;FPP SY:[204,6]R4SORT                                               22
;FPP SY:[204,6]R4TYPREP                                             23
;                                                                   24
;   ROUTINES WITH PREFIX R5                                         25
;                                                                   26
;FPP SY:[204,6]R5CHNGES                                             27
;FPP SY:[204,6]R5CTYPE                                              28
;                                                                   29
;   ROUTINES WITH PREFIX UT                                         30
;                                                                   31
;FPP SY:[204,7]UTDRDCIF                                             32
;FPP SY:[204,7]UTMAKNAM                                             33
;FPP SY:[204,7]UTSQEEZ                                              34
;                                                                   35
;   COMPILE FORTRAN ROUTINES                                        36
;                                                                   37
@[204,6]R4FOR.CMD                                                   38
;                                                                   39
;   @R4FOR.CMD                                                      40
;                                                                   41
;   COMPILE FORTRAN ROUTINES FOR THE COMPONENT INFORMATION REPORT BY  42
;   TYPE PROGRAM (REP4)   (P. LO    5/26/82)                        43
;                                                                   44
;   ROUTINES WITH PREFIX R4                                         45
;                                                                   46
;FOR/F4P/OBJECT:[204,6]R4DCTION [204,6]R4DCTION                     47
;FOR/F4P/OBJECT:[204,6]R4PERCNT [204,6]R4PERCNT                     48
;FOR/F4P/OBJECT:[204,6]R4REP4   [204,6]R4REP4                       49
;FOR/F4P/OBJECT:[204,6]R4SORT   [204,6]R4SORT                       50
;FOR/F4P/OBJECT:[204,6]R4TYPREP [204,6]R4TYPREP                     51
;                                                                   52
;   ROUTINES WITH PREFIX R5                                         53
;                                                                   54
;FOR/F4P/OBJECT:[204,6]R5CHNGES [204,6]R5CHNGES                     55
```

Figure 3-16. REP4 Task Generation Command Procedure (R4GEN.CMD) (1 of 2)

```
;FOR/F4P/OBJECT:[204,6]R5CTYPE  [204,6]R5CTYPE                                  56
;                                                                              57
;   ROUTINES WITH PREFIX UT                                                    58
;                                                                              59
;FOR/F4P/OBJECT:[204,7]UTDRDCIF [204,7]UTDRDCIF                                 60
;FOR/F4P/OBJECT:[204,7]UTMAKNAM [204,7]UTMAKNAM                                 61
;FOR/F4P/OBJECT:[204,7]UTSQEEZ  [204,7]UTSQEEZ                                  62
;                                                                              63
;   BUILD THE REP4 TASK IMAGE                                                  64
;                                                                              65
TKB @[204,6]R4.TKB                                                             66
;                                                                              67
;   @R4.TKB                                                                    68
;                                                                              69
;   CIF TYPE AND COMPLEXITY REPORT PROGRAM (REP4) OVERLAY     DEC 79           70
;                                                                              71
;[204,5]R4/FU,R4=[204,6]R4/MP                                                   72
;ACTFIL=3                                                                       73
;UNITS=20                                                                       74
;ASG=SY:2:6:11,TI:5                                                             75
://                                                                            76
```

Figure 3-16. REP4 Task Generation Command Procedure (R4GEN.CMD) (2 of 2)

The CG program is generated by entering the following command:

```
@[204,6]CGGEN
```

The REP4 program is generated by entering this command:

```
@[204,6]R4GEN
```

3.5.4.2 Overlay Structure

The CG and REP4 programs are both overlaid to reduce the memory space requirement. The files containing the Overlay Descriptor Language needed to generate the task images for these two programs are [204,6]CG.ODL and [204,6]R4.ODL, respectively. Figure 3-17 is a listing of CG.ODL; Figure 3-18 is R4.ODL. The system libraries RMS11M.ODL and RMS12X.ODL are needed for both overlays. In addition, the RMS Indexed Access Program Library (RMSIAC) is needed in both overlays. The name of the library is [204,7]UFRMSIAC.OLB. It contains the FORTRAN routines necessary to access RMS indexed files.

```
;                                                                          1
;   @CG.ODL                                                                2
;                                                                          3
;   THE CHANGE AND ERROR ACCUMULATION PROGRAM (CG) OVERLAY                 4
;   (P. LO    5/20/82)                                                     5
;                                                                          6
        .ROOT   $TREE1,OTSALL,RMSALL                                       7
$TREE1: .FCTR   [204,6]CGXCH-RMSROT-OTSROT-[204,7]UFRMSIAC/LB-*($LV)       8
$LV:    .FCTR   [204,7]UTMAKNAM,$CHDA,[204,7]UTSQEEZ                       9
$CHDA:  .FCTR   [204,6]CGCHDATA-$C1-*([204,7]UTDRDCRF,[204,6]CGFILEIT)     10
$C1:    .FCTR   [204,7]UFRMSIAC/LB                                         11
;                                                                          12
;                                                                          13
@LB:[1,1]RMS11M.ODL                                                        14
@LB:[1,1]RMS12X.ODL                                                        15
        .END                                                               16
```

Figure 3-17. CG Program Overlay Descriptor Language File (CG.ODL)

```
;                                                                          1
;   @R4.ODL                                                                2
;                                                                          3
;  OVERLAY STRUCTURE FOR THE COMPONENT INFORMATION REPORT BY               4
;   FUNCTION PROGRAM (REP4)                                                5
;   (P. LO    5/5/82)                                                      6
;                                                                          7
        .ROOT   $TREE1,OTSALL,RMSALL                                       8
$TREE1: .FCTR   [204,6]R4REP4-RMSROT-OTSROT-$R1                            9
$R1:    .FCTR   [204,7]UTDRDCIF-[204,7]UFRMSIAC/LB-$FORT                   10
$FORT:  .FCTR   *([204,7]UTMAKNAM,[204,6]R5CTYPE,$SORT,$DIC,$SQ,$TYPE)     11
$SORT:  .FCTR   [204,6]R4SORT                                              12
$DIC:   .FCTR   [204,6]R4DCTION                                            13
$SQ:    .FCTR   [204,7]UTSQEEZ                                             14
$TYPE:  .FCTR   [204,6]R4TYPREP-*([204,6]R5CHNGES,[204,6]R4PERCNT)         15
;                                                                          16
;                                                                          17
@LB:[1,1]RMS11M.ODL                                                        18
@LB:[1,1]RMS12X.ODL                                                        19
        .END                                                               20
```

Figure 3-18. REP4 Program Overlay Descriptor Language File (R4.ODL)

3.6 COMPONENT INFORMATION REPORT PROGRAM (REP5)

3.6.1 INTRODUCTION

The Component Information Report Program (REP5) produces a list of components and associated data for a given project. For each component, basic data from the Component Information File (CIF), Halstead parameters computed from the basic data, and the change and error data retrieved from the CG intermediate file produced by the CG program (Section 3.5) are reported. Correlation coefficients between the various statistics presented are also given.

3.6.2 PROGRAM STRUCTURE

3.6.2.1 Files Accessed

The REP5 program accesses two input files, one output file, and one scratch file, as described below.

Input File Name	Description
[204,1]<PRJNAM>.CIF	CIF for the given project
<PRJNAM>.CHN	CG intermediate file containing change and error data produced by the CG program for the given project

Output File Name	Description
<PRJNAM>.RP5	REP5 output report for the given project

A scratch file is used by the REP5 program to temporarily store data that will later be used to compute the correlation coefficient matrix.

For these file names, <PRJNAM> is the name of the project selected by the user.

3.6.2.2 Baseline Diagram

Figure 3-19 is the baseline diagram for the REP5 program. The REP5 routine is the driver that obtains the project name and selected subsystem, reads the CG intermediate file and

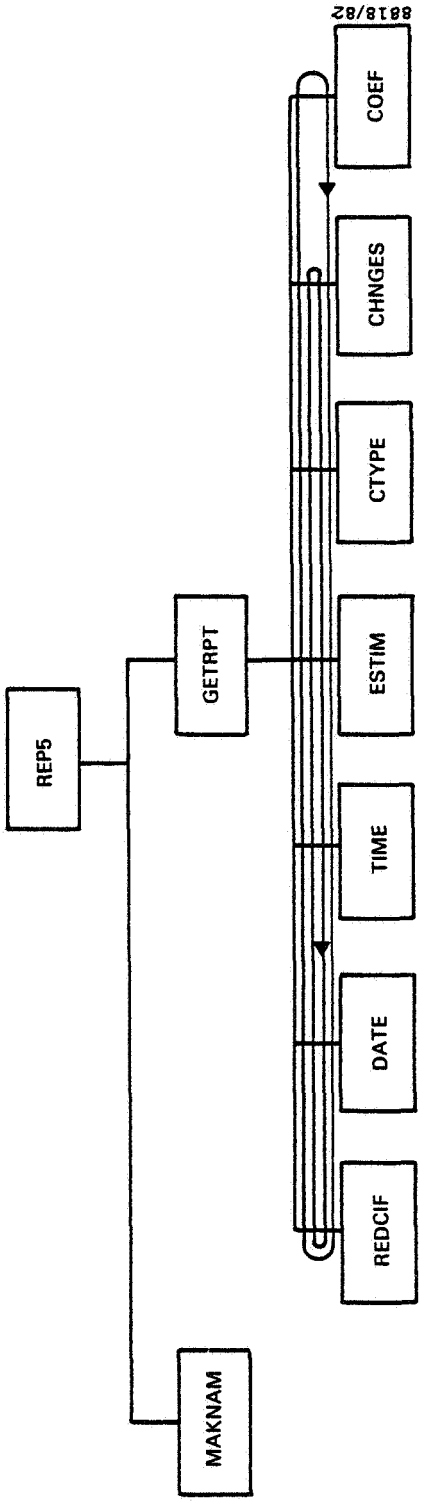


Figure 3-19. Baseline Diagram for the Component Information Report Program (REP5)

CIF for the given project, computes the Halstead parameters, and writes the output report. REP5 loops through the above process until a^Z (control Z) is returned in response to a subsystem prefix prompt.

3.6.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines referenced by the REP5 program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major REP5 routines are described in Section 3.6.3.4. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines in this section, the REP5 program also uses the following system routines: DATE, SECNDS, and TIME.

3.6.3.1 Process Data and Compute Statistics

These five major routines obtain data from a given CIF and compute statistics for the output report.

ROUTINE: COEF

FUNCTION: Computes the correlation coefficient matrix for a given set of variables

CALLING SEQUENCE:

```
CALL COEF (ISCRAH, IREPF, NUM, IREC, TITLE)
```

ROUTINE: CTYPE

FUNCTION: Determines the function type of a component

CALLING SEQUENCE:

```
CALL CTYPE (ICTEXC, ICTFNR, ICTIO, KASGN, KCALL, KFMT,
            ITYPE)
```

ROUTINE: ESTIM

FUNCTION: Computes the values of several Halstead parameters

CALLING SEQUENCE:

```
CALL ESTIM (ICTHIO, IETA1, IETA2, NETA1, NETA2,
            IETA, NETA, LENGTH, IVOL, PRGLVL, ALNGLV,
            IEFORT, TOTIM, NBUGS, IVSTAR, STROUD, ERROR)
```

ROUTINE: GETRPT

FUNCTION: Extracts pertinent data from the CIF and writes it to the output report

CALLING SEQUENCE:

```
CALL GETRPT (LUNDB, ITERMF, IREPF, ISCRAH, ICHNGF,
             PROJNM)
```

ROUTINE: REP5

FUNCTION: Main routine of the REP5 program, extracts data from the CIF and from the CG intermediate file, computes statistics, and writes the output report

CALLING SEQUENCE: None

3.6.3.2 File Open and Read Routines

These two routines either open an indexed file or read records from a file.

ROUTINE: CHNGES

FUNCTION: Reads the CG intermediate file and returns the number of changes and errors for a given component name; if the component name is not found, the routine returns 999 for the output variables

CALLING SEQUENCE:

```
CALL CHNGES (ANAME, ICHNGF,
             NCHS, NERRS, TOTCH)
```

ROUTINE: REDCIF

FUNCTION: Reads one record from the CIF and converts all data to internal format

CALLING SEQUENCE:

```
CALL REDCIF (LUNDB,
             PROJNO, CNAME, ICODE, PANV, MODFUN, SYSFUN,
             ORIGIN, NEXEC, NLINES, NCOMNT, IETA1, IETA2,
             NETA1, NETA2, NIOVAR, MCCABE, NFUNCT, NIO,
             NASGN, NCALL, NFMT, EOF, ERROR)
```

3.6.3.3 Routine Performing String Operations

This routine performs a string operation.

ROUTINE: MAKNAM

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL MAKNAM (DISK, UIC, NAME, EXTENS,
             DSN)
```

3.6.3.4 Variable Description

The variables in the calling sequences of major REP5 routines are described below.

Name	Type	Description
ALNGLV	R*4	Language level
ANAME	R*8	Component name
ERROR	L*1	Error flag
ICHNGF	I*2	Change and error data file (CG intermediate file) unit number
ICTEXC	I*2	Number of executable statements
ICTFNR	I*2	Number of function references
ICTHIO	I*2	Number of input and output variables for component

Name	Type	Description
ICTIO	I*2	Number of I/O statements
IEFORT	I*4	Effort required
IETA	I*2	Number of unique elements
IETA1	I*2	Number of unique operators
IETA2	I*2	Number of unique operands
IREC	I*2	Total number of records in file
IREPF	I*2	Unit number associated with the REP5 output report file
ISCRAH	I*2	Unit number associated with the scratch file
ITERMF	I*2	Unit number associated with the terminal
ITYPE	I*2	Component function type
IVOL	I*2	Program volume
IVSTAR	I*2	Potential volume
KASGN	I*2	Number of assignment statements
KCALL	I*2	Number of CALLs
KFMT	I*2	Number of FORMAT statements
LENGTH	I*2	Predicted length
LUNDB	I*2	Unit number associated with the CIF
NBUGS	I*2	Predicted number of bugs
NCHS	I*2	Number of changes for the given component
NERRS	I*2	Number of errors
NETA	I*2	Total number of elements
NETA1	I*2	Total number of operators
NETA2	I*2	Total number of operands
NUM	I*2	Number of lines of data
PRGLVL	R*4	Program level
PROJNM(8)	L*1	Project name
STROUD	I*4	Stroud number (discriminations per hour)
TITLE(10)	R*8	Arrays of column titles

Name	Type	Description
TOTCH	I*2	Total number of changes and errors
TOTIM	R*4	Total programming time required

3.6.4 TASK BUILD PROCEDURE

3.6.4.1 Command Procedures

The REP5 program can be generated from the source code by executing the command procedure R5GEN.CMD under UIC [204,6]. This command procedure references three command procedures--R5FPP.CMD, R5FOR.CMD, and R5.TKB--all under UIC [204,6]. Figure 3-20 is a listing of R5GEN.CMD, the command procedure to precompile, compile, and task build the REP5 program. The REP5 program is generated by entering the following command:

```
@[204,6]R5GEN
```

3.6.4.2 Overlay Structure

The REP5 program is overlaid to reduce the memory space requirement. Figure 3-21 is a listing of the Overlay Descriptor Language file, [204,6]R5.ODL, needed to build the REP5 program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @R5GEN.CMD                                                             2
;                                                                          3
;   COMMAND PROCEDURE TO PRECOMPILE, COMPILE AND TASK BUILD THE REP5       4
;   PROGRAM     (P. LO      6/14/82)                                       5
;                                                                          6
;   PRECOMPILE FORTRAN ROUTINES                                            7
;                                                                          8
@[204,6]R5FPP.CMD                                                          9
;                                                                         10
;   @R5FPP.CMD                                                            11
;                                                                         12
;   COMMAND PROCEDURE TO PRECOMPILE FORTRAN ROUTINES FOR REP5 PROGRAM     13
;   (P. LO      6/14/82)                                                  14
;                                                                         15
;FPP SY:[204,6]R5CHNGES                                                   16
;FPP SY:[204,6]R5COEF                                                     17
;FPP SY:[204,6]R5CTYPE                                                    18
;FPP SY:[204,6]R5ESTIM                                                    19
;FPP SY:[204,6]R5GETRPT                                                   20
;FPP SY:[204,6]R5REP5                                                     21
;                                                                         22
;FPP SY:[204,7]UTMAKNAM                                                   23
;FPP SY:[204,7]UTREDCIF                                                   24
;                                                                         25
;   COMPILE FORTRAN ROUTINES                                              26
;                                                                         27
@[204,6]R5FOR.CMD                                                         28
;                                                                         29
;   @R5FOR.CMD                                                            30
;                                                                         31
;   COMMAND PROCEDURE TO COMPILE ALL FORTRAN ROUTINES FOR THE REP5        32
;   PROGRAM   (P. LO    6/14/82)                                          33
;                                                                         34
;FOR/F4P/OBJECT:[204,6]R5CHNGES [204,6]R5CHNGES                           35
;FOR/F4P/OBJECT:[204,6]R5COEF   [204,6]R5COEF                             36
;FOR/F4P/OBJECT:[204,6]R5CTYPE  [204,6]R5CTYPE                            37
;FOR/F4P/OBJECT:[204,6]R5ESTIM  [204,6]R5ESTIM                            38
;FOR/F4P/OBJECT:[204,6]R5GETRPT [204,6]R5GETRPT                           39
;FOR/F4P/OBJECT:[204,6]R5REP5   [204,6]R5REP5                             40
;                                                                         41
;FOR/F4P/OBJECT:[204,7]UTMAKNAM [204,7]UTMAKNAM                           42
;FOR/F4P/OBJECT:[204,7]UTREDCIF [204,7]UTREDCIF                           43
;                                                                         44
;   TASK BUILD THE REP5 PROGRAM                                           45
;                                                                         46
TKB @[204,6]R5.TKB                                                        47
;                                                                         48
;   @R5.TKB                                                               49
;                                                                         50
;   COMMAND PROCEDURE TO TASK BUILD THE COMPONENT INFORMATION REPORT       51
;   PROGRAM (REP5)                                                        52
;                                                                         53
;[204,5]R5=[204,6]R5.ODL/MP                                               54
```

Figure 3-20. Task Generation Command Procedure for the REP5 Program (R5GEN.CMD)

```
;                                                                    1
;   @R5.ODL                                                          2
;                                                                    3
;   OVERLAY STRUCTURE FOR THE COMPONENT INFORMATION REPORT PROGRAM    4
;   (REP5)   (P. LO     6/14/82)                                     5
;                                                                    6
          .ROOT $ROOT,OTSALL,RMSALL                                  7
$ROOT:    .FCTR [204,6]R5REP5-RMSROT-OTSROT-*(NAME,FORT)             8
NAME:     .FCTR [204,7]UTMAKNAM                                      9
FORT:     .FCTR [204,6]R5GETRPT-*(RCIF,COEF,CHNGS,EST,TYPE)         10
RCIF:     .FCTR [204,7]UTREDCIF                                      11
COEF:     .FCTR [204,6]R5COEF                                        12
CHNGS:    .FCTR [204,6]R5CHNGES                                      13
EST:      .FCTR [204,6]R5ESTIM                                       14
TYPE:     .FCTR [204,6]R5CTYPE                                       15
;                                                                    16
;                                                                    17
@LB:[1,1]RMS11M                                                      18
@LB:[1,1]RMS12X                                                      19
          .END                                                       20
```

Figure 3-21. REP5 Program Overlay Descriptor Language File (R5.ODL)

3.7 GRAPHING PROGRAM (GQ)

3.7.1 INTRODUCTION

The Graphing Program (GQ) reads an external data file containing a set of points and produces a graph of the data. It also optionally fits a polynomial of degree less than or equal to 10 to the given set of points and computes various associated statistics.

3.7.2 PROGRAM STRUCTURE

3.7.2.1 Files Accessed

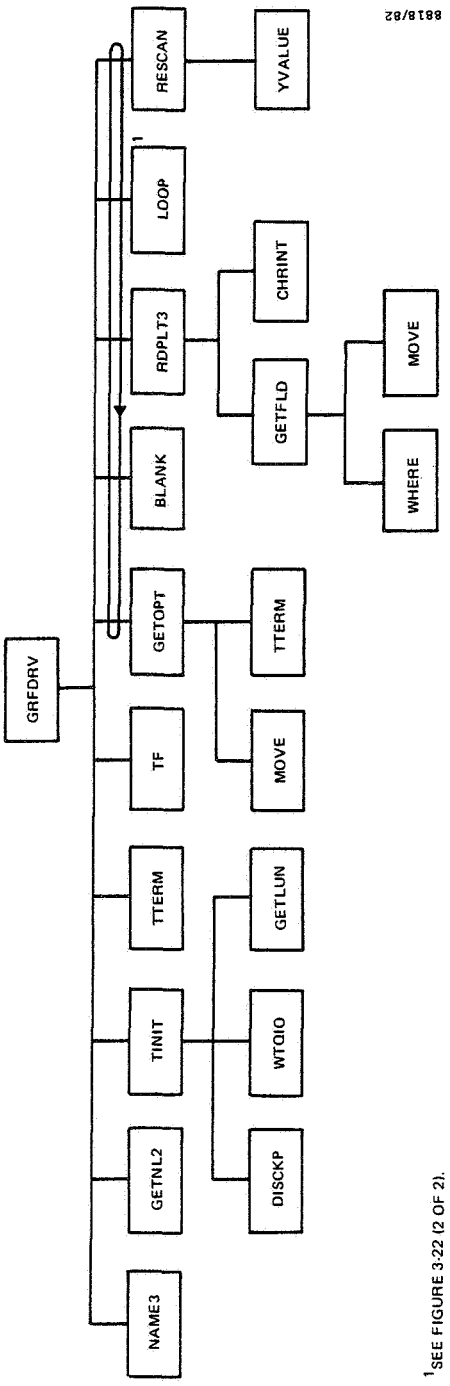
The GQ program accesses two input files and one output file as described below.

Input File Name	Description
[204,6]GQ.NL	GQ input parameters file
<PRJNAM>.XXX	External file containing project name, X-axis title, Y-axis title, and a set of X, Y values for the points to be plotted. The file name for the external data file is of the form <PRJNAM>.XXX if produced by the PF or the WK program, where <PRJNAM> is the name of the project for which the program was executed and XXX denotes the type of data (Sections 2.2.3 and 2.4.3). If generated by the user, the file name is arbitrary.

Output File Name	Description
FOR0XX.DAT	Output graph and statistics report (XX is the output unit number specified in the GQ input parameters file).

3.7.2.2 Baseline Diagram

Figure 3-22 is the baseline diagram for the GQ program. The GRFDRV routine is the main driver. It reads the GQ input parameters file, initializes the user's terminal, reads the external data file, and produces a graph of the given data.



[1] SEE FIGURE 3-22 (2 OF 2).

Figure 3-22. Baseline Diagram for the Graphing Program (GQ) (1 of 2)

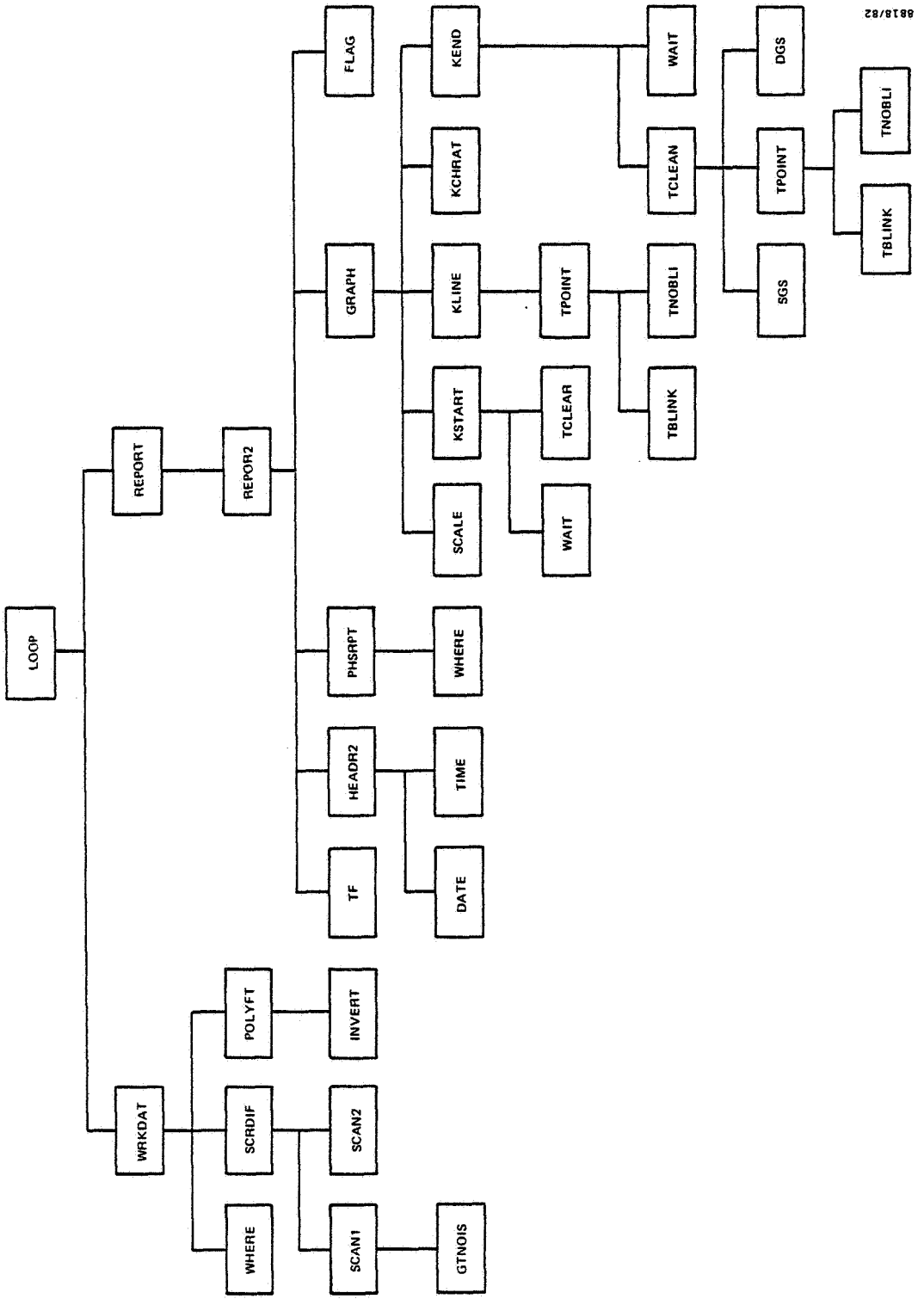


Figure 3-22. Baseline Diagram for the Graphing Program (GQ) (2 of 2)

3.7.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The subroutines forming the GQ program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major GQ routines are described in Section 3.7.3.6. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the GQ program also uses the following system routines: CLEAR, DATE, DISCKP, GETLUN, TIME, WAIT, and WTQIO.

3.7.3.1 Process Data and Compute Statistics

These 13 major routines obtain data from the external data file, compute statistics, and produce the graph.

ROUTINE: FLAG (LOGICAL FUNCTION)

FUNCTION: Sets a given character to the flag character if the given character is blank

CALLING SEQUENCE:

```
FLAG (CHAR, QFLAG)
```

ROUTINE: GRFDRV

FUNCTION: Main routine of the GQ program, reads a file containing a set of points and produces a graph of the data

CALLING SEQUENCE: None

ROUTINE: GTNOIS

FUNCTION: Computes a noise value from the data points

CALLING SEQUENCE:

```
CALL GTNOIS (NPTS, QFLAG, Y,
             QCHARS,
             AVNOIS, ERROR)
```

ROUTINE: INVERT

FUNCTION: Inverts a matrix in place and solves a set of simultaneous linear equations

CALLING SEQUENCE:

```
CALL INVERT (A, B, N, L,
             C, IER)
```

ROUTINE: LOOP

FUNCTION: Computes the minimum chi square and rejects data points outside a specified factor times the standard deviation; also prints a graph and statistics as desired

CALLING SEQUENCE:

```
CALL LOOP (ADEBUG, DDATE, DIFFAC, IOFFSE, IOP, IPAGE,
           IPART1, IPART4, IPR, IWID, KCYCLE, KXSHFT,
           MCOIN, MLINES, MXFRAC, MXITER, MXORDR, NAV1,
           NAV2, NPTS, NSTREK, PROJ, QBAND, QBEST,
           QCHARS, QCHR, QCUM, QCYCLE, QFLAG, QGRAPH,
           QINTG, QMARKR, QNL, QOMITO, QPRINT, QRESCN,
           QSCALX, QSCREN, QSTATS, QTRUNC, RES, RPTITL,
           SIGFAC, TOL, X, XFACTR, XH, XTITLE, Y, YDFAC,
           YFACTR, YH, YLOW, YTITLE,
           CHIONE,
           COEF, MCO, STDV)
```

ROUTINE: POLYFT

FUNCTION: Performs a least-squares polynomial fit to a set of data points

CALLING SEQUENCE:

```
CALL POLYFT (X, Y, NPTS, MCOEF, TOL, QFLAG, QCHARS,
             CHI, COEF, RES, STDV, SUMABS, SUMMR2, SUMR2,
             XMEAN, IER)
```

ROUTINE: RESCAN

FUNCTION: Checks to determine if points should be flagged or unflagged

CALLING SEQUENCE:

```
CALL RESCAN (COEF, MCO, NPTS, QCHR1, QFLAG, RES, STDV,
             X, Y,
             QCHARS)
```

ROUTINE: SCAN1

FUNCTION: Performs a preliminary scan on the data and flags those points obviously out of a reasonable range

CALLING SEQUENCE:

```
CALL SCAN1 (DIFFAC, NPTS, QCHR1, QFLAG, Y, NSTREK,
            QCHARS,
            AVNOIS, ERROR)
```

ROUTINE: SCAN2

FUNCTION: Cycles through all points (ignoring previously flagged points) and computes the average Y-values for the previous NPTS points and the succeeding NPTS points; flags the current point if the difference between its Y-value and these averages exceeds a specified tolerance

CALLING SEQUENCE:

```
CALL SCAN2 (AVNOIS, MXFRAC, MXITER, NAV1, NAV2, NPTS,
            NSTREK, QCHR1, QFLAG, Y, YDFAC,
            QCHARS,
            YDFAC2, ERROR)
```

ROUTINE: SCRDIF

FUNCTION: Computes the average difference in Y-values for all data points and flags data points whose difference from the previous point and subsequent point varies more than a given factor times the average difference

CALLING SEQUENCE:

```
CALL SCRDIF (DIFFAC, MXFRAC, MXITER, NAV1, NAV2, NPTS,
             NSTREK, QCHR1, QFLAG, Y, YDFAC,
             QCHARS,
             AVNOIS, YDFAC2, ERROR)
```

ROUTINE: TF (LOGICAL FUNCTION)

FUNCTION: Returns a value of .TRUE. if the input number is not zero

CALLING SEQUENCE:

```
TF (N)
```

ROUTINE: WRKDAT

FUNCTION: Takes the given X and Y arrays and manipulates and scales the data as desired by the given input parameters; also computes several statistics related to the standard deviation

CALLING SEQUENCE:

```
CALL WRKDAT  (ADEBUG, CHIONE, DIFFAC, IOFFSE, IPART4,
              MCO, MXFRAC, MXITER, NAV1, NAV2, NSTREK,
              QBAND, QCHR, QFLAG, QOMIT0, QSCALX, QSCREN,
              QTRUNC, SIGFAC, TOL, YDFAC, YFACTR,
              NPTS, QCHARS, QMARKR, X, Y,
              AFRAC, AREA1, AREA2, AVNOIS, CHI, COEF,
              KZEROS, NPTPLT, NPTREJ, RES, STDV, SUMABS,
              SUMMR2, SUMR2, XFACTR, XMEAN, YDFAC2, ERROR)
```

ROUTINE: YVALUE (REAL FUNCTION)

FUNCTION: Computes the Y-value associated with a given X-value for the polynomial with the given coefficients and degree

CALLING SEQUENCE:

```
YVALUE (COEF, MCO, XVAL)
```

3.7.3.2 Print a Graph and Statistics Report

These five routines produce a graph and statistics chart of the given data.

ROUTINE: GRAPH

FUNCTION: Generates a one-page Cartesian printer plot for any set of data with automatic scaling

CALLING SEQUENCE:

```
CALL GRAPH (IOPT, IPR, IWID, KXSHFT, MLINES, N, N2,
           QCHARS, QMARKR, QXTITL, QYTITL, X, XH, XL, Y,
           YH, YL,)
           LINES)
```

ROUTINE: HEADR2

FUNCTION: Prints a one-line title for each report page that includes the date and project name

CALLING SEQUENCE:

```
CALL HEADR2 (IRPTF, PRJNAM, RPTITL,
             IPAGE)
```

ROUTINE: PHSRPT

FUNCTION: Prints phase date information on the first page of the graphing report

CALLING SEQUENCE:

```
CALL PHSRPT (IPR, NPTS, QMARKR, T10)
```

ROUTINE: REPORT

FUNCTION: Produces a graph and statistical chart of the given data

CALLING SEQUENCE:

```
CALL REPORT (QBAND, QBEST, QCUM, QCYCLE, QGRAPH, QINTG,
             QNL, QOMIT0, QPRINT, QSCREN, QSTATS, QTRUNC,
             QRESCN)
```

ROUTINE: REPOR2

FUNCTION: An ENTRY point of routine REPORT

CALLING SEQUENCE:

```
CALL REPOR2 (AFRAC, AREA1, AREA2, AVNOIS, CHI, COEF,
             DDATE, DIFFAC, IOFFSE, IOPT, IPAGE, IPART1,
             IPART4, IPR, IRES, IWID, KCYCLE, KXSHFT,
             KZEROS, MCO, MLINES, MXFRAC, MXITER, MXORDR,
             NAV1, NAV2, NPTPLT, NPTREJ, NPTS, NSTREK,
             PROJ, QCHARS, QFLAG, QMARKR, RES, RPTITL,
             SIGFAC, STDV, SUMABS, SUMMR2, SUMR2, TOL, X,
             XFACTR, XH, XMEAN, XTITLE, Y, YDFAC, YDFAC2,
             YFACTR, YH, YLOW, YTITLE)
```

3.7.3.3 Obtain Data From Terminal or External Data Set

These four routines obtain information from a user's response to a terminal prompt or from an external data set.

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR,
             FIELD)
```

ROUTINE: GETNL2

FUNCTION: Reads a sequential file and fills a parameter array

CALLING SEQUENCE:

```
CALL GETNL2 (NLDSN, NLFIL, MAXNL,
             NL, ERROR)
```

ROUTINE: GETOPT

FUNCTION: Retrieves user options for the current run

CALLING SEQUENCE:

```
CALL GETOPT (IPR, IWID, MCO, MLINES, QCHR, QDSN, QEOF)
```

ROUTINE: RDPLT3

FUNCTION: Reads an external data file for X and Y values and X and Y axis titles

CALLING SEQUENCE:

```
CALL RDPLT3 (IPLTF, PLTNAM, EXTFIL, MAXREC, QCUM, QMAKEX,
             TERMNL,
             CHAR, PROJ, RPTITL, PIETTL, X, Y, NCOUNT,
             XHI, XTITLE, YHI, YTITLE, MARKER, DDATE,
             FACTRY, EOFTTY, ERROR)
```

3.7.3.4 Routines for String Movement or Comparison

These five routines deal with string movement or comparison.

ROUTINE: BLANK

FUNCTION: Initializes an array to blanks

CALLING SEQUENCE:

```
CALL BLANK (ARRAY, NUM)
```

ROUTINE: CHRINT

FUNCTION: Converts the given string to integers in I*2 format

CALLING SEQUENCE:

```
CALL CHRINT (CHARS, NCHAR,
             I2NUM, ERROR)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

ROUTINE: WHERE

FUNCTION: Locates the given characters in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
            LOC, FOUND)
```

3.7.3.5 Plot Routines

These 12 routines deal with plotting the graph on the terminal or graphing equipment.

ROUTINE: KCHRAT (LOGICAL FUNCTION)

FUNCTION: Obtains the character at the given point

CALLING SEQUENCE:

```
KCHRAT (X, Y)
```

ROUTINE: KEND

FUNCTION: Finishes production of a graph and prints the developed grid

CALLING SEQUENCE:

```
CALL KEND (LINES)
```

ROUTINE: KLINE

FUNCTION: Writes the given character string to the current file, terminal, or IIS graphics device

CALLING SEQUENCE:

```
CALL KLINE (X, Y, DIR, LEN, CHARS)
```

ROUTINE: KSTART

FUNCTION: Initializes the screen or IIS graphics device and a grid for a plot

CALLING SEQUENCE:

```
CALL KSTART
```

ROUTINE: SCALE

FUNCTION: Chooses the best scale for plotting any set of data

CALLING SEQUENCE:

```
CALL SCALE (XMIN, XMAX, NMAX,
            XI, DX, NX, NDECX, NDIGX)
```

ROUTINE: TBLINK

FUNCTION: Turns on the blink function of the VT100 terminal

CALLING SEQUENCE:

```
CALL TBLINK
```

ROUTINE: TCLEAN

FUNCTION: Finishes the production of a graph and prints the developed grid

CALLING SEQUENCE:

```
CALL TCLEAN (QGRID, XMAX, YMAX, LINES)
```

ROUTINE: TCLEAR

FUNCTION: Clears the terminal or IIS graphics device

CALLING SEQUENCE:

```
CALL TCLEAR
```

ROUTINE: TINIT

FUNCTION: Initializes the terminal in preparation for graphics

CALLING SEQUENCE:

```
CALL TINIT
```

ROUTINE: TNOBLI

FUNCTION: Turns off the blink option of the VT100 terminal

CALLING SEQUENCE:

```
CALL TNOBLI
```

ROUTINE: TPOINT

FUNCTION: Writes the given characters starting at the given point

CALLING SEQUENCE:

```
CALL TPOINT (X, Y, DIR, LEN, CHARS)
```

ROUTINE: TTERM

FUNCTION: Changes the default terminal number

CALLING SEQUENCE:

```
CALL TTERM (JTERM)
```

3.7.3.6 Variable Description

The variables in the calling sequences of major GQ routines are described below.

Name	Type	Description
ADEBUG(80)	I*2	Debug array
AFRAC	R*4	Area under computed curve divided by area under actual data
AREA1	R*4	Area under computed curve
AREA2	R*4	Area under actual data (including flagged points)
AVNOIS	R*4	Average noise value
CHAR	L*1	A given character
CHI	R*8	Chi square
CHIONE	L*1	Flag indicating if first attempt to fit polynomial
COEF(10)	R*8	Coefficients of fit
DDATE(9)	L*1	Date of data
DIFFAC	R*4	Difference factor
ERROR	L*1	Error flag
IOFFSE	I*2	Parameter that forces start and end of curve fit to data to 0, if 1; if 0, does not force curve to 0

Name	Type	Description
IOPT	I*2	Parameter to plot count of overlapping points, if 1; if 0, does not plot count
IPAGE	I*2	Current page number
IPART1	I*2	Maximum number of points allowed
IPART4	I*2	Size of X, Y, and character arrays (4 * IPART1)
IPR	I*2	Output unit number
IRES	I*2	Number of reject cycle
IWID	I*2	Width of graph in columns, including titles
KCYCLE	I*2	Number of times to cycle through data rejecting flagged points
KXSHFT	I*2	Column to start graph
KZEROS	I*2	Number of trailing zero data points flagged
MCO	I*2	Order of fit desired
MCOIN	I*2	Minimum order of polynomial to be fit to data
MLINES	I*2	Number of rows allowed in graph
MXFRAC	R*4	Maximum fraction of flagged points
MXITER	I*2	Maximum number of iterations
MXORDR	I*2	Maximum order of polynomial to be fit to data
NAV1	I*2	Number of preceding points to consider
NAV2	I*2	Number of succeeding points to consider
NPTPLT	I*2	Number of points plotted
NPTREJ	I*2	Number of points flagged (rejected)
NPTS	I*2	Number of data points
NSTREK	I*2	Maximum number of consecutive flagged points allowed
PROJ(8)	L*1	Project name
QBAND	L*1	Flag indicating whether to plot band around fitted curve
QBEST	L*1	Flag indicating whether program is to find polynomial of best fit
QCHARS (NPTS)	L*1	Array of characters to be plotted

Name	Type	Description
QCHR(4)	L*1	Characters to be used: = 1, Data points = 2, Upper edge of band around curve = 3, Lower edge of band around curve = 4, Curve fit to data points
QCHR1	L*1	Data point character for unflagged points
QCUM	L*1	Flag indicating whether to accumulate data as it is read in.
QCYCLE	L*1	Flag indicating whether to print graph report each time through reject cycle
QDSN(27)	L*1	Name of file to be read
QEOF	L*1	End of file flag
QFLAG	L*1	Flag character
QGRAPH	L*1	Flag indicating whether to print graph page
QINTG	L*1	Flag indicating whether to print data as integers on last page of report
QMARKR (IPART1)	L*1	Array of characters to be printed at bottom of graph (phase characters)
QNL	L*1	Flag indicating whether to print input parameter (first) page of report
QOMIT0	L*1	Flag indicating whether to ignore zero data points
QPRINT	L*1	Flag indicating whether to print graph report each cycle through curve fitting process
QRESCN	L*1	Flag indicating whether to recheck editing of data and fitting of polynomial
QSCALX	L*1	Flag indicating whether to scale X data points
QSCREN	L*1	Flag indicating whether to screen data points relative to preceeding and succeeding points
QSTATS	L*1	Flag indicating whether to print statistics page of report
QTRUNC	L*1	Flag indicating whether to truncate zero data points at end of array
RES(NPTS)	R*4	Residuals from curve fit to data

Name	Type	Description
RPTITL(40)	L*1	Report title
SIGFAC	R*4	Sigma factor used to plot band around curve fit to data
STDV	R*8	Standard deviation
SUMABS	R*8	Sum of absolute residuals
SUMMR2	R*8	Sum of minimum residuals squared
SUMR2	R*8	Sum of residuals squared
TOL	R*4	Tolerance of data
T10	I*2	Tab location of printed information
X(NPTS)	R*4	X data values
XFACTR	R*4	X scaling factor
XH	R*4	X axis maximum
XMEAN	R*8	Mean Y value
XTITLE(40)	L*1	X axis title
XVAL	R*4	X value
Y(NPTS)	R*4	Y data values
YDFAC	R*4	Y delta factor
YDFAC2	R*4	Final prescan boundary factor
YFACTR	R*4	Y scaling factor
YH	R*4	Y axis maximum
YLOW	R*4	Y axis minimum
YTITLE(40)	L*1	Y axis title

3.7.4 TASK BUILD PROCEDURE

3.7.4.1 Command Procedures

The GQ task can be generated from the source code by executing the command procedure GQGEN.CMD under UIC [204,6]. This command procedure references three command files--GQFPP.CMD, GQFOR.CMD, and GQ.TKB--all under UIC [204,6]. Figure 3-23 is a listing of GQGEN.CMD, the command procedure to precompile, compile, and task build the GQ program. The GQ program is generated by entering the following command:

```
@[204,6]GQGEN
```

```
;                                                                          1
;    @GQGEN.CMD                                                            2
;                                                                          3
;    COMMAND PROCEDURE TO TASK BUILD THE GRAPHING PROGRAM (GQ) FROM        4
;    FORTRAN SOURCE    (P. LO    7/8/82)                                   5
;                                                                          6
;    PRECOMPILE FORTRAN SOURCE                                             7
;                                                                          8
@[204,6]GQFPP.CMD                                                          9
;                                                                         10
;    @GQFPP.CMD                                                           11
;                                                                         12
;    COMMAND PROCEDURE TO PRECOMPILE FORTRAN ROUTINES FOR THE GRAPHING     13
;    PROGRAM (GQ)    (P. LO    7/6/82)                                    14
;                                                                         15
;    ROUTINES WITH PREFIX GQ                                              16
;                                                                         17
;FPP SY:[204,6]GQFLAG                                                     18
;FPP SY:[204,6]GQGETOPT                                                   19
;FPP SY:[204,6]GQGRFDRV                                                   20
;FPP SY:[204,6]GQGTNOIS                                                   21
;FPP SY:[204,6]GQLOOP                                                     22
;FPP SY:[204,6]GQPHSRPT                                                   23
;FPP SY:[204,6]GQREPORT                                                   24
;FPP SY:[204,6]GQRESCAN                                                   25
;FPP SY:[204,6]GQSCAN1                                                    26
;FPP SY:[204,6]GQSCAN2                                                    27
;FPP SY:[204,6]GQSCRDIF                                                   28
;FPP SY:[204,6]GQWRKDAT                                                   29
;FPP SY:[204,6]GQYVALUE                                                   30
;                                                                         31
;    ROUTINES WITH PREFIX SK                                              32
;                                                                         33
;FPP SY:[204,7]SKKCHRAT                                                   34
;FPP SY:[204,7]SKKEND                                                     35
;FPP SY:[204,7]SKKLINE                                                    36
;FPP SY:[204,7]SKKSTART                                                   37
;                                                                         38
;    ROUTINES WITH PREFIX ST                                              39
;                                                                         40
;FPP SY:[204,7]STTBLINK                                                   41
;FPP SY:[204,7]STTCLEAN                                                   42
;FPP SY:[204,7]STTCLEAR                                                   43
;FPP SY:[204,7]STTINIT                                                    44
;FPP SY:[204,7]STTNOBLI                                                   45
;FPP SY:[204,7]STTPOINT                                                   46
;FPP SY:[204,7]STTTERM                                                    47
;                                                                         48
;    ROUTINES WITH PREFIX UT                                              49
;                                                                         50
;FPP SY:[204,7]UTBLANK                                                    51
;FPP SY:[204,7]UTCHRINT                                                   52
;FPP SY:[204,7]UTGETFLD                                                   53
;FPP SY:[204,7]UTGETNL2                                                   54
;FPP SY:[204,7]UTGRAPH                                                    55
```

Figure 3-23. GQ Task Generation Command Procedure (GQGEN.CMD) (1 of 3)

```
;FPP SY:[204,7]UTHEADR2
;FPP SY:[204,7]UTINVERT
;FPP SY:[204,7]UTMOVE
;FPP SY:[204,7]UTNAME3
;FPP SY:[204,7]UTPOLYFT
;FPP SY:[204,7]UTRDPLT3
;FPP SY:[204,7]UTSCALE
;FPP SY:[204,7]UTTF
;FPP SY:[204,7]UTWHERE
;
;   COMPILE FORTRAN SOURCE
;
@[204,6]GQFOR.CMD
;
;   @GQFOR.CMD
;
;   COMMAND PROCEDURE TO COMPILE FORTRAN ROUTINES FOR THE GRAPHING
;   PROGRAM (GQ)    (P. LO    7/6/82)
;
;   ROUTINES WITH PREFIX GQ
;
;FOR/F4P/OBJECT:[204,6]GQFLAG   [204,6]GQFLAG
;FOR/F4P/OBJECT:[204,6]GQGETOPT [204,6]GQGETOPT
;FOR/F4P/OBJECT:[204,6]GQGRFDRV [204,6]GQGRFDRV
;FOR/F4P/OBJECT:[204,6]GQGTNOIS [204,6]GQGTNOIS
;FOR/F4P/OBJECT:[204,6]GQLOOP   [204,6]GQLOOP
;FOR/F4P/OBJECT:[204,6]GQPHSRPT [204,6]GQPHSRPT
;FOR/F4P/OBJECT:[204,6]GQREPORT [204,6]GQREPORT
;FOR/F4P/OBJECT:[204,6]GQRESCAN [204,6]GQRESCAN
;FOR/F4P/OBJECT:[204,6]GQSCAN1  [204,6]GQSCAN1
;FOR/F4P/OBJECT:[204,6]GQSCAN2  [204,6]GQSCAN2
;FOR/F4P/OBJECT:[204,6]GQSCRDIF [204,6]GQSCRDIF
;FOR/F4P/OBJECT:[204,6]GQWRKDAT [204,6]GQWRKDAT
;FOR/F4P/OBJECT:[204,6]GQYVALUE [204,6]GQYVALUE
;
;   ROUTINES WITH PREFIX SK
;
;FOR/F4P/OBJECT:[204,7]SKKCHRAT [204,7]SKKCHRAT
;FOR/F4P/OBJECT:[204,7]SKKEND   [204,7]SKKEND
;FOR/F4P/OBJECT:[204,7]SKKLINE  [204,7]SKKLINE
;FOR/F4P/OBJECT:[204,7]SKKSTART [204,7]SKKSTART
;
;   ROUTINES WITH PREFIX ST
;
;FOR/F4P/OBJECT:[204,7]STTBLINK [204,7]STTBLINK
;FOR/F4P/OBJECT:[204,7]STTCLEAN [204,7]STTCLEAN
;FOR/F4P/OBJECT:[204,7]STTCLEAR [204,7]STTCLEAR
;FOR/F4P/OBJECT:[204,7]STTINIT  [204,7]STTINIT
;FOR/F4P/OBJECT:[204,7]STTNOBLI [204,7]STTNOBLI
;FOR/F4P/OBJECT:[204,7]STTPOINT [204,7]STTPOINT
;FOR/F4P/OBJECT:[204,7]STTTERM  [204,7]STTTERM
;
;   ROUTINES WITH PREFIX UT
;
;FOR/F4P/OBJECT:[204,7]UTBLANK  [204,7]UTBLANK
```

Figure 3-23. GQ Task Generation Command Procedure (GQGEN.CMD) (2 of 3)

```
;FOR/F4P/OBJECT:[204,7]UTCHRINT [204,7]UTCHRINT                          111
;FOR/F4P/OBJECT:[204,7]UTGETFLD [204,7]UTGETFLD                          112
;FOR/F4P/OBJECT:[204,7]UTGETNL2 [204,7]UTGETNL2                          113
;FOR/F4P/OBJECT:[204,7]UTGRAPH  [204,7]UTGRAPH                           114
;FOR/F4P/OBJECT:[204,7]UTHEADR2 [204,7]UTHEADR2                          115
;FOR/F4P/OBJECT:[204,7]UTINVERT [204,7]UTINVERT                          116
;FOR/F4P/OBJECT:[204,7]UTMOVE   [204,7]UTMOVE                            117
;FOR/F4P/OBJECT:[204,7]UTNAME3  [204,7]UTNAME3                           118
;FOR/F4P/OBJECT:[204,7]UTPOLYFT [204,7]UTPOLYFT                          119
;FOR/F4P/OBJECT:[204,7]UTRDPLT3 [204,7]UTRDPLT3                          120
;FOR/F4P/OBJECT:[204,7]UTSCALE  [204,7]UTSCALE                           121
;FOR/F4P/OBJECT:[204,7]UTTF     [204,7]UTTF                              122
;FOR/F4P/OBJECT:[204,7]UTWHERE  [204,7]UTWHERE                           123
;                                                                       124
;  COMPILE ASSEMBLER ROUTINE                                            125
;                                                                       126
MAC/OBJECT:[204,7]VT [204,7]VT                                          127
;                                                                       128
;   BUILD THE GQ TASK                                                   129
;                                                                       130
TKB @[204,6]GQ.TKB                                                      131
;                                                                       132
;   @GQ.TKB                                                             133
;                                                                       134
;   COMMAND PROCEDURE TO TASK BUILD THE GRAPHING PROGRAM (GQ)           135
;                                                                       136
;[204,5]GQ=[204,6]GQ/MP                                                 137
;UNITS=25                                                               138
```

Figure 3-23. GQ Task Generation Command Procedure (GQGEN.CMD) (3 of 3)

3.7.4.2 Overlay Structure

The GQ program is overlaid to reduce the memory space requirement. Figure 3-24 is a listing of the Overlay Descriptor Language file, [204,6]GQ.ODL, needed to build the GQ program task image.

```
;                                                                          1
;   $GQ.ODL                                                                2
;                                                                          3
;   OVERLAY DEFINITION FOR THE GRAPHING PROGRAM (GQ)                        4
;                                                                          5
        .ROOT $ROOT-*($NL,$OPT,$LOOP-($WRK,$RPT),$RESC)                    6
$ROOT:  .FCTR [204,6]GQGRFDRV-[204,7]UTTF     -$ROT1                       7
$ROT1:  .FCTR [204,6]GQFLAG  -$ROT6                                        8
$ROT6:  .FCTR [204,7]STTBLINK-[204,7]STTCLEAN-[204,7]STTCLEAR-$ROT7        9
$ROT7:  .FCTR [204,7]STTINIT -[204,7]STTNOBLI-[204,7]STTPOINT-$ROT8       10
$ROT8:  .FCTR [204,7]STTTERM -$ROT9                                       11
$ROT9:  .FCTR $ROT10                                                      12
$ROT10: .FCTR [204,7]UTMOVE  -[204,7]UTNAME3 -[204,7]UTBLANK -$ROT11      13
$ROT11: .FCTR [204,7]UTWHERE -[204,7]UTGETFLD-$ROT12                      14
$ROT12: .FCTR [204,7]UTCHRINT-$ROT13                                      15
$ROT13: .FCTR [204,7]UTRDPLT3                                             16
;                                                                         17
$NL:    .FCTR [204,7]UTGETNL2                                             18
;                                                                         19
$OPT:   .FCTR [204,6]GQGETOPT                                             20
;                                                                         21
$LOOP:  .FCTR [204,6]GQLOOP                                               22
;                                                                         23
$WRK:   .FCTR [204,6]GQWRKDAT-($WA,$WB)                                   24
$WA:    .FCTR [204,7]UTPOLYFT-[204,7]UTINVERT                             25
$WB:    .FCTR [204,6]GQSCRDIF-($WC,$WD)                                   26
$WC:    .FCTR [204,6]GQSCAN1 -[204,6]GQGTNOIS                             27
$WD:    .FCTR [204,6]GQSCAN2                                              28
;                                                                         29
$RPT:   .FCTR [204,6]GQREPORT-($RA,$RB,$RC)                               30
$RA:    .FCTR [204,7]UTHEADR2                                             31
$RB:    .FCTR [204,6]GQPHSRPT                                             32
$RC:    .FCTR [204,7]UTGRAPH -[204,7]UTSCALE -$RC2                        33
$RC2:   .FCTR [204,7]VT      -$RC3                                        34
$RC3:   .FCTR [204,7]SKKSTART-[204,7]SKKLINE -[204,7]SKKCHRAT-$RC4        35
$RC4:   .FCTR [204,7]SKKEND                                               36
;                                                                         37
$RESC:  .FCTR [204,6]GQRESCAN-[204,6]GQYVALUE                             38
;                                                                         39
        .END                                                              40
```

Figure 3-24. GQ Program Overlay Descriptor Language File (GQ.ODL)

3.8 FORM COUNTER PROGRAM (NF)

3.8.1 INTRODUCTION

The Form Counter Program (NF) produces a report containing counts of forms in the SEL data base files for a given project. Counts are given by form type and programmer for the following types of forms: Change Report Form (CRF), Component Summary Form (CSF), Component Status Report (CSR), Run Analysis Form (RAF), and Resource Summary Form (RSF).

3.8.2 PROGRAM STRUCTURE

3.8.2.1 Files Accessed

The NF program accesses eight input files and one output file as described below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) File
[204,1]EST.HDR	EST file
[204,1]HEADER.HDR	HDR file
[204,1]<PRJNAM>.CRF	CRF file for the given project
[204,1]<PRJNAM>.CSF	CSF file for the given project
[204,1]<PRJNAM>.CSR	CSR file for the given project
[204,1]<PRJNAM>.RAF	RAF file for the given project
[204,1]<PRJNAM>.RSF	RSF file for the given project

Output File Name	Description
<PRJNAM>.NF	Output report for the given project

In these file names, <PRJNAM> is the name of the project selected by the user.

3.8.2.2 Baseline Diagram

Figure 3-25 is the baseline diagram for the NF program. The NFORMS routine is the main driver. It obtains the project name; counts all forms on the CRF, CSF, CSR, RAF, and RSF files; and then produces a report of form counts for the given project.

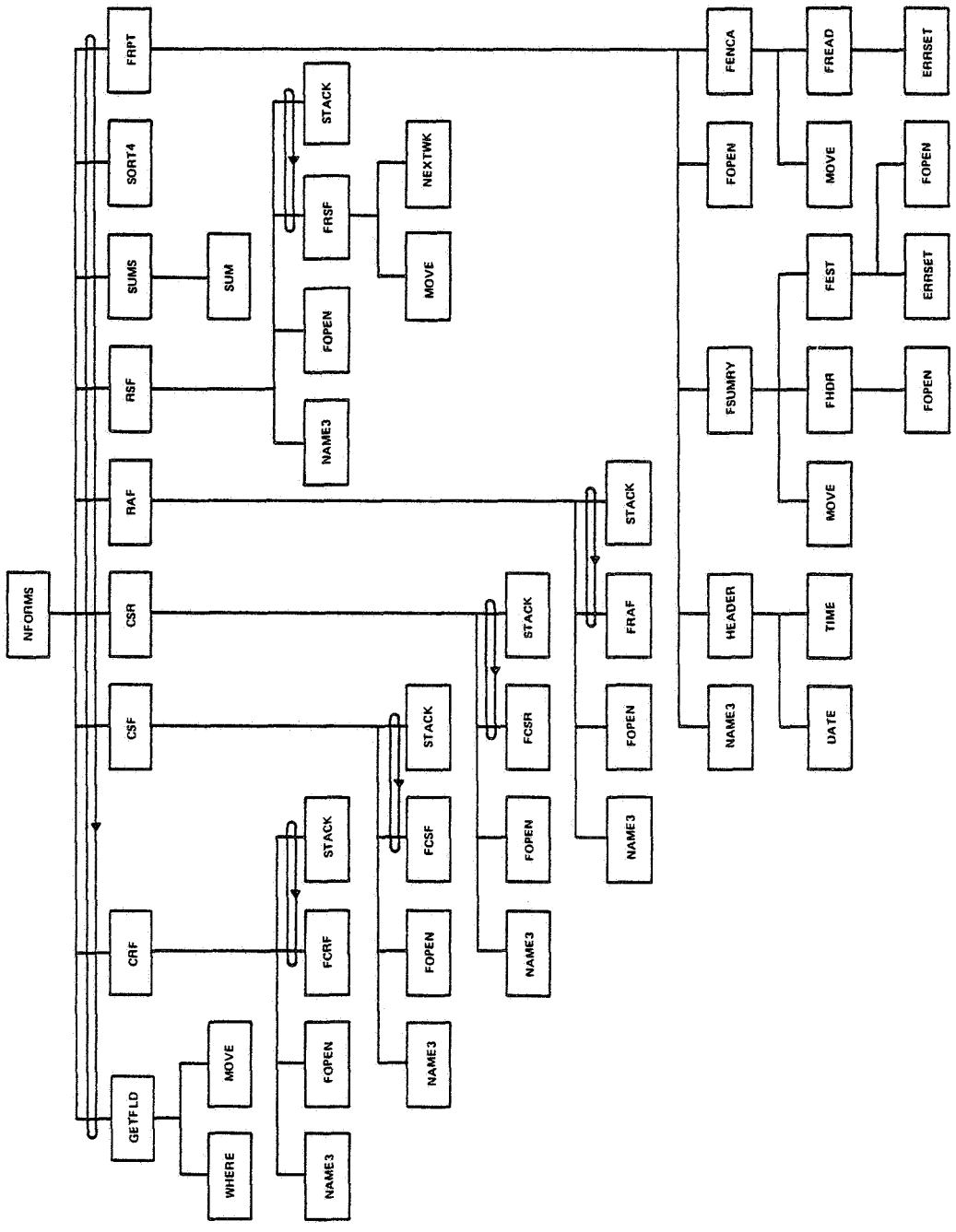


Figure 3-25. Baseline Diagram for the Form Counter Program (NF)

3.8.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the NF program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major NF routines are described in Section 3.8.3.8. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the NF program also uses the following system routines: DATE, ERRSET, SECNDS, and TIME.

3.8.3.1 Process Data and Count Forms

These seven major routines count all forms on the data base in each file for a given project.

ROUTINE: CRF

FUNCTION: Totals all CRFs

CALLING SEQUENCE:

```
CALL CRF (PROJCT,
          ERROR, NPROG, PROG,
          NCRF)
```

ROUTINE: CSF

FUNCTION: Totals all CSFs

CALLING SEQUENCE:

```
CALL CSF (PROJCT,
          ERROR, NPROG, PROG,
          NCSF)
```

ROUTINE: CSR

FUNCTION: Totals all CSRs

CALLING SEQUENCE:

```
    CALL CSR (PROJCT,
              ERROR, NPROG, PROG,
              NCSR)
```

ROUTINE: NFORMS

FUNCTION: Main routine of the NF program, counts all forms on the data base in each file for the given project

CALLING SEQUENCE: None

ROUTINE: RAF

FUNCTION: Totals all RAFs

CALLING SEQUENCE:

```
    CALL RAF (PROJCT,
              ERROR, NPROG, PROG,
              NRAF)
```

ROUTINE: RSF

FUNCTION: Totals all RSFs

CALLING SEQUENCE:

```
    CALL RSF (PROJCT,
              ERROR, NPROG, PROG,
              NRSF)
```

ROUTINE: SUMS

FUNCTION: Totals all form counts

CALLING SEQUENCE:

```
    CALL SUMS (NPROG, NATM, NCRF, NCSF, NCSR, NGPS, NRAF,
               NRSF,
               NALL, TATM, TCRF, TCSF, TCSR, TGPS, TRAF,
               TRSF, TALL)
```

3.8.3.2 Write the Form Count Report

These three routines write the report of form counts for the given project.

ROUTINE: FRPT

FUNCTION: Prints a report of form counts of each form type by programmer

CALLING SEQUENCE:

```
CALL FRPT (NATM, NCRF, NCSF, NCSR, NGPS, NRAF, NRSF,
          NALL, TATM, TCRF, TCSF, TCSR, TGPS, TRAF,
          TRSF, TALL, IORDER, NPROG, PROG, PROJCT)
```

ROUTINE: FSUMRY

FUNCTION: Prints a six-line header summary with data from the HDR and EST files

CALLING SEQUENCE:

```
CALL FSUMRY (IRPTF, PRJNAM)
```

ROUTINE: HEADER

FUNCTION: Prints a one-line title for each report page including the date and project name

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PRJNAM, RPTITL)
```

3.8.3.3 Obtain Data From Terminal or External Data Set

These two routines obtain information from a user's response to a terminal prompt or from an external data set.

ROUTINE: FENCA

FUNCTION: Locates the description field on the Encoding Dictionary corresponding to the given type and code

CALLING SEQUENCE:

```
CALL FENCA (IENCF, TYPE, CODE,
            NAME, REST, FOUND)
```

ROUTINE: GETFLD

FUNCTION: Displays the given text on the terminal and prompts for a given character string

CALLING SEQUENCE:

```
CALL GETFLD (TEXT, EXTFIL, FLDLEN,
             TERMNL, EOFTTY, ERROR,
             FIELD)
```

3.8.3.4 Sort and Search Routines

These two routines provide sort and search functions.

ROUTINE: SORT4

FUNCTION: Produces an array of indices sorted according to the given I*4 array

CALLING SEQUENCE:

```
CALL SORT4 (I4, NSORT,
            ISORT)
```

ROUTINE: STACK

FUNCTION: Determines if the given name is in the current list and adds it if it is not

CALLING SEQUENCE:

```
CALL STACK (MX, PROGNO,
            NPROG, PROG,
            NFRM, ERROR)
```

3.8.3.5 File Open and Read Routines

These nine routines either open an indexed file or read records from an indexed file.

ROUTINE: FCRF

FUNCTION: Reads one record from the CRF file and converts the data to internal format

CALLING SEQUENCE:

```
CALL FCRF (ICRFF,
          FORMNO, PROJNO, PROGNO, FDATE, NCH, NEXAM,
          OVER1, DATDET, DATBEG, EFFORT, CHTYPE, CHCOMP,
          ERRTYP, ERRIN, DATERR, LGCERR, ACTVTY, ISOLTM,
          PATCH, RELOLD, RELNO, RELDAT, CMTREA, CMTDES,
          CMTGEN, STATUS, EOF, ERROR)
```

ROUTINE: FCSF

FUNCTION: Reads one record from the CSF file

CALLING SEQUENCE:

```
CALL FCSF (ICSFF,
          FORMNO, PROJNO, PROGNO, PROGI, FDATE, FSTAGE,
          COMPCO, PRECIS, CMPLEX, SWTYPE, PASGN, PCNTL,
          POTHER, STATWO, STMT, BTSIZE, INDEP, RELSW,
          ADDTYP, NCALLD, X1, NCALNG, X2, NSHR, X3,
          NDESC, X4, LANG1, PLANG1, LANG2, PLANG2, DES,
          CONSTR, DESRUN, CODRUN, TSTRUN, DESTIM,
          CODTIM, TSTTIM, DESEFF, CODEFF, TSTEFF,
          DESDAT, CODDAT, TSTDAT, DESCR, CALLD, CALNG,
          SHR, AFFECT, OTH, NAMCON, CMT1, CMT2, ISTAT,
          EOF, ERROR)
```

ROUTINE: FCSR

FUNCTION: Reads one record from the CSR file using FORTRAN read

CALLING SEQUENCE:

```
CALL FCSR (ICSRF,
          FORMNO, SEQNO, PROJNO, PROGNO, FDATE, COMPCO,
          TIMES, OTHNAM, OTHOUR, ISTAT, PHASE, EOF,
          ERROR)
```

ROUTINE: FEST

FUNCTION: Reads one record from the EST file and converts all data to internal format

CALLING SEQUENCE:

```
CALL FEST (IESTF, NAME,
          PROJ, NCOMP, MODDEL, MODNEW, MODMOD, NRUNS,
          NCHANG, PAGDOC, LINDEL, LINNEW, LINMOD,
          TOTEXT, NEWEXT, MODEXT, PROGHR, MGMTHR,
          OTHRHR, HR95, HR75, OTHCMP, STATUS, ACTIVE,
          PRJCAT, FOUND, ERROR)
```

ROUTINE: FHDR

FUNCTION: Reads one record from the HDR file using the secondary key (project name)

CALLING SEQUENCE:

```
CALL FHDR (IHDRF, PRJNAM,
          PROJ, DEVCMP, TARG, ALIEN, RANGES, STATUS,
          ERROR)
```

ROUTINE: FOPEN

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL FOPEN (IUNIT, FILNAM,
           ERROR)
```

ROUTINE: FRAF

FUNCTION: Reads one record from the RAF file using FORTRAN read

CALLING SEQUENCE:

```
CALL FRAF (IRAFF,
          FORMNO, SEQNO, PROJNO, PROGNO, RDATE, MACHIN,
          INTERA, PURPOS, NCOMP, COMPCO, FIRST, METOBJ,
          RESULT, COMENT, ISTAT, EOF, ERROR)
```

ROUTINE: FREAD

FUNCTION: Reads one indexed record

CALLING SEQUENCE:

```
CALL FREAD (IUNIT, KEYVAL, KEYLEN, LRECL,
           BUFFER, ERROR)
```

ROUTINE: FRSF

FUNCTION: Reads one record from the RSF file using FORTRAN read; returns all data on that record converted to internal format plus an array containing each week for which there are data in the record

CALLING SEQUENCE:

```
CALL FRSF (IRSFF,
          FORMNO, SEQNO, PROJNO, RESCOD, RESID, FDATE,
          PCMGMT, WKDATE, NRUNS, TIMES, STATUS, PHASE,
          LASTWK, EOF, ERROR)
```

3.8.3.6 Routines for String Movement or Comparison

These three routines deal with string movement or comparison.

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
           DSN)
```

ROUTINE: WHERE

FUNCTION: Locates the given characters in the given string

CALLING SEQUENCE:

```
CALL WHERE (CHAR, STRING, LEN,
           LOC, FOUND)
```

3.8.3.7 Mathematical Functions

These two routines perform mathematical functions.

ROUTINE: NEXTWK

FUNCTION: Computes the date 1 week after the given date and returns it in MM, DD, YY format

CALLING SEQUENCE:

```
CALL NEXTWK (DATE,
             D)
```

ROUTINE: SUM (INTEGER*2 FUNCTION)

FUNCTION: Computes the sum of all integers in a given array

CALLING SEQUENCE:

```
SUM (ARRAY, N)
```

3.8.3.8 Variable Description

The variables in the calling sequences of major NF routines are described below.

Name	Type	Description
ARRAY(MX)	I*2	Array of numbers
ERROR	L*1	Error flag
IORDER(MX)	I*2	Sorted index array for programmer numbers
ISORT (NSORT)	I*2	Sorted index array
I4(NSORT)	I*4	Array on which sort is based
MX	I*2	Number of programmers allowed
N	I*2	Number of array elements to be summed
NALL(MX)	I*2	Number of all forms for each programmer
NATM(MX)	I*2	Number of Attitude Maintenance (ATM) forms for each programmer
NCRF(MX)	I*2	Number of CRFs for each programmer
NCSF(MX)	I*2	Number of CSFs for each programmer
NCSR(MX)	I*2	Number of CSRs for each programmer
NFRM(MX)	I*2	Number of forms for each programmer for given form type
NGPS(MX)	I*2	Number of General Project Summary (GPS) forms for each programmer
NPROG	I*2	Number of programmers found
NRAF(MX)	I*2	Number of RAFs for each programmer
NRSF(MX)	I*2	Number of RSFs for each programmer
NSORT	I*2	Number of entries in array I4
PROG(MX)	I*4	Array of programmer numbers
PROGNO	I*4	Given programmer number
PROJCT(8)	L*1	Project name
TALL	I*2	Total number of all forms
TATM	I*2	Total number of ATM forms
TCRF	I*2	Total number of CRFs
TCSF	I*2	Total number of CSFs
TCSR	I*2	Total number of CSRs

Name	Type	Description
TGPS	I*2	Total number of GPS forms
TRAF	I*2	Total number of RAFs
TRSF	I*2	Total number of RSFs

3.8.4 TASK BUILD PROCEDURE

3.8.4.1 Command Procedures

The NF program can be generated from the source code by executing the command procedure NFGEN.CMD under UIC [204,6]. This command procedure references three command files--NFFPP.CMD, NFFOR.CMD, and NF.TKB--all under UIC [204,6]. Figure 3-26 is a listing of NFGEN.CMD, the command procedure to precompile, compile, and task build the NF program. The NF program is generated by executing the following command:

```
@[204,6]NFGEN
```

3.8.4.2 Overlay Structure

The NF program is overlaid to reduce the memory space requirement. Figure 3-27 is a listing of the Overlay Descriptor Language file, [204,6]NF.ODL, needed to build the NF program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @NFGEN.CMD                                                             2
;                                                                          3
;   COMMAND PROCEDURE TO GENERATE THE FORM COUNTER PROGRAM (NF) FROM       4
;   THE STRUCTURED FORTRAM SOURCE CODES                                    5
;                                                                          6
;   PRECOMPILE FORTRAN ROUTINES                                            7
;                                                                          8
@[204,6]NFFPP.CMD                                                          9
;                                                                         10
;   @NFFPP.CMD                                                            11
;                                                                         12
;   COMMAND PROCEDURE TO PRECOMPILE ALL FORTRAN ROUTINES FOR THE FORM     13
;   COUNTER PROGRAM (NF)   (P. LO   7/14/82)                              14
;                                                                         15
;   ROUTINES WITH PREFIX NF                                               16
;                                                                         17
;FPP SY:[204,6]NFCRF                                                      18
;FPP SY:[204,6]NFCSF                                                      19
;FPP SY:[204,6]NFCSR                                                      20
;FPP SY:[204,6]NFFRPT                                                     21
;FPP SY:[204,6]NFNFORMS                                                   22
;FPP SY:[204,6]NFRAF                                                      23
;FPP SY:[204,6]NFRSF                                                      24
;FPP SY:[204,6]NFSORT                                                     25
;FPP SY:[204,6]NFSTACK                                                    26
;FPP SY:[204,6]NFSUM                                                      27
;FPP SY:[204,6]NFSUMS                                                     28
;                                                                         29
;   ROUTINES WITH PREFIX UT                                               30
;                                                                         31
;FPP SY:[204,7]UTFCRF                                                     32
;FPP SY:[204,7]UTFCSF                                                     33
;FPP SY:[204,7]UTFCSR                                                     34
;FPP SY:[204,7]UTFENCA                                                    35
;FPP SY:[204,7]UTFEST                                                     36
;FPP SY:[204,7]UTFHDR                                                     37
;FPP SY:[204,7]UTFOPEN                                                    38
;FPP SY:[204,7]UTFRAF                                                     39
;FPP SY:[204,7]UTFREAD                                                    40
;FPP SY:[204,7]UTFRSF                                                     41
;FPP SY:[204,7]UTFSUMRY                                                   42
;FPP SY:[204,7]UTGETFLD                                                   43
;FPP SY:[204,7]UTHEADER                                                   44
;FPP SY:[204,7]UTMOVE                                                     45
;FPP SY:[204,7]UTNAME3                                                    46
;FPP SY:[204,7]UTNEXTWK                                                   47
;FPP SY:[204,7]UTWHERE                                                    48
;                                                                         49
;   COMPILE FORTRAN ROUTINES                                              50
;                                                                         51
@[204,6]NFFOR.CMD                                                         52
;                                                                         53
;   @NFFOR.CMD                                                            54
;                                                                         55
```

Figure 3-26. NF Task Generation Command Procedure (NFGEN.CMD) (1 of 2)

```
:   COMMAND PROCEDURE TO COMPILE FORTRAN ROUTINES FOR THE FORM          56
:   COUNTER PROGRAM (NF)   (P. LO    7/15/82)                          57
:                                                                      58
:   ROUTINES WITH PREFIX NF                                            59
:                                                                      60
;FOR/F4P/OBJECT:[204,6]NFCRF     [204,6]NFCRF                          61
;FOR/F4P/OBJECT:[204,6]NFCSF     [204,6]NFCSF                          62
;FOR/F4P/OBJECT:[204,6]NFCSR     [204,6]NFCSR                          63
;FOR/F4P/OBJECT:[204,6]NFFRPT    [204,6]NFFRPT                         64
;FOR/F4P/OBJECT:[204,6]NFNFORMS  [204,6]NFNFORMS                       65
;FOR/F4P/OBJECT:[204,6]NFRAF     [204,6]NFRAF                          66
;FOR/F4P/OBJECT:[204,6]NFRSF     [204,6]NFRSF                          67
;FOR/F4P/OBJECT:[204,6]NFSORT    [204,6]NFSORT                         68
;FOR/F4P/OBJECT:[204,6]NFSTACK   [204,6]NFSTACK                        69
;FOR/F4P/OBJECT:[204,6]NFSUM     [204,6]NFSUM                          70
;FOR/F4P/OBJECT:[204,6]NFSUMS    [204,6]NFSUMS                         71
:                                                                      72
:   ROUTINES WITH PREFIX UT                                            73
:                                                                      74
;FOR/F4P/OBJECT:[204,7]UTFCRF    [204,7]UTFCRF                         75
;FOR/F4P/OBJECT:[204,7]UTFCSF    [204,7]UTFCSF                         76
;FOR/F4P/OBJECT:[204,7]UTFCSR    [204,7]UTFCSR                         77
;FOR/F4P/OBJECT:[204,7]UTFENCA   [204,7]UTFENCA                        78
;FOR/F4P/OBJECT:[204,7]UTFEST    [204,7]UTFEST                         79
;FOR/F4P/OBJECT:[204,7]UTFHDR    [204,7]UTFHDR                         80
;FOR/F4P/OBJECT:[204,7]UTFOPEN   [204,7]UTFOPEN                        81
;FOR/F4P/OBJECT:[204,7]UTFRAF    [204,7]UTFRAF                         82
;FOR/F4P/OBJECT:[204,7]UTFREAD   [204,7]UTFREAD                        83
;FOR/F4P/OBJECT:[204,7]UTFRSF    [204,7]UTFRSF                         84
;FOR/F4P/OBJECT:[204,7]UTFSUMRY  [204,7]UTFSUMRY                       85
;FOR/F4P/OBJECT:[204,7]UTGETFLD  [204,7]UTGETFLD                       86
;FOR/F4P/OBJECT:[204,7]UTHEADER  [204,7]UTHEADER                       87
;FOR/F4P/OBJECT:[204,7]UTMOVE    [204,7]UTMOVE                         88
;FOR/F4P/OBJECT:[204,7]UTNAME3   [204,7]UTNAME3                        89
;FOR/F4P/OBJECT:[204,7]UTNEXTWK  [204,7]UTNEXTWK                       90
;FOR/F4P/OBJECT:[204,7]UTWHERE   [204,7]UTWHERE                        91
:                                                                      92
:   TASK BUILD THE NF PROGRAM                                          93
:                                                                      94
TKB @[204,6]NF.TKB                                                     95
:                                                                      96
:   @NF.TKB                                                            97
:                                                                      98
:   COMMAND PROCEDURE TO TASK BUILD THE FORM COUNTER PROGRAM (NF)      99
:                                                                     100
;[204,5]NF=[204,6]NF/MP                                               101
;MAXBUF=250                                                           102
://                                                                   103
```

Figure 3-26. NF Task Generation Command Procedure (NFGEN.CMD) (2 of 2)

```
;                                                                          1
;   @NF.ODL                                                                2
;                                                                          3
;   OVERLAY DESCRIPTOR LANGUAGE FOR THE FORM COUNTER PROGRAM (NF)          4
;                                                                          5
        .ROOT   $ROOT,OTSALL,RMSALL                                        6
$ROOT:  .FCTR   $R1-$R2-$R5-$R6-RMSROT-OTSROT-$SUBS                        7
$R1:    .FCTR   [204,6]NFNFORMS-[204,6]NFSTACK -[204,6]NFSUM               8
$R2:    .FCTR   [204,6]NFSORT  -[204,7]UTGETFLD-[204,7]UTWHERE             9
$R5:    .FCTR   [204,7]UTNAME3 -[204,7]UTMOVE                             10
$R6:    .FCTR   [204,7]UTFREAD -[204,7]UTFOPEN                            11
;                                                                         12
$SUBS:  .FCTR   *($CRF,$CSF,$CSR,$RAF,$RSF,$FRPT,$SUMS)                   13
$CRF:   .FCTR   [204,6]NFCRF    -[204,7]UTFCRF                            14
$CSF:   .FCTR   [204,6]NFCSF    -[204,7]UTFCSF                            15
$CSR:   .FCTR   [204,6]NFCSR    -[204,7]UTFCSR                            16
$RAF:   .FCTR   [204,6]NFRAF    -[204,7]UTFRAF                            17
$RSF:   .FCTR   [204,6]NFRSF    -[204,7]UTFRSF-[204,7]UTNEXTWK            18
$FRPT:  .FCTR   [204,6]NFFRPT   -($HDR,$FSUM,$ENC)                        19
$HDR:   .FCTR   [204,7]UTHEADER                                           20
$FSUM:  .FCTR   [204,7]UTFSUMRY-($HED,$EST)                               21
$HED:   .FCTR   [204,7]UTFHDR                                             22
$EST:   .FCTR   [204,7]UTFEST                                             23
$ENC:   .FCTR   [204,7]UTFENCA                                            24
$SUMS:  .FCTR   [204,6]NFSUMS                                             25
;                                                                         26
@LB:[1,1]RMS11M.ODL                                                       27
@LB:[1,1]RMS12X.ODL                                                       28
        .END                                                              29
```

Figure 3-27. NF Program Overlay Descriptor Language File (NF.ODL)

3.9 SEL DATA BASE LISTING PROGRAM (LISTDB)

3.9.1 INTRODUCTION

The SEL Data Base Listing Program (LISTDB) produces formatted and interpreted listings of SEL data base files. File types include Attitude Maintenance (ATM), Component Information File (CIF), Change Report Form (CRF), Component Summary Form (CSF), Component Status Report (CSR), Growth History (HIS), Run Analysis Form (RAF), and Resource Summary Form (RSF).

3.9.2 PROGRAM STRUCTURE

3.9.2.1 Files Accessed

The LISTDB program accesses nine input files and eleven output files as described below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file
[204,1]<PRJNAM>.CIF	CIF for the given project
[204,1]<PRJNAM>.CRF	CRF file for the given project
[204,1]<PRJNAM>.CSF	CSF file for the given project
[204,1]<PRJNAM>.CSR	CSR file for the given project
[204,1]<PRJNAM>.HIS	HIS file for the given project
[204,1]<PRJNAM>.RAF	RAF file for the given project
[204,1]<PRJNAM>.RSF	RSF file for the given project
[204,1]<PRJNAM>.ATM	ATM file for the given project

In these file names,<PRJNAM> denotes the name of the project selected by the user.

Output File Name	Description
LISTDB.CIF	Output listing of the CIF
LISTDB.CRF	Output listing of the CRF file (change report)
LISTDB.ERR	Output listing of the CRF file (error report)
LISTDB.CF1	Output listing of the CSF file (part one)

Output File Name	Description
LISTDB.CF2	Output listing of the CSF file (part two)
LISTDB.CF3	Output listing of the CSF file (part three)
LISTDB.CSR	Output listing of the CSR file
LISTDB.HIS	Output listing of the HIS file
LISTDB.RAF	Output listing of the RAF file
LISTDB.RSF	Output listing of the RSF file
LISTDB.ATM	Output listing of the ATM file

3.9.2.2 Baseline Diagram

Figure 3-28 is the baseline diagram for the LISTDB program. The LISTDB routine is the main driver. It obtains the project names and file types and then processes the selected data base files and displays them.

3.9.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the LISTDB program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major LISTDB routines are described in Section 3.9.3.5. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the LISTDB program also uses the following system routines: ERRSET and SECNDS.

3.9.3.1 Process Data and Produce Formatted Lists of Files

These 21 major routines process data and produce a formatted list of an SEL data base file.

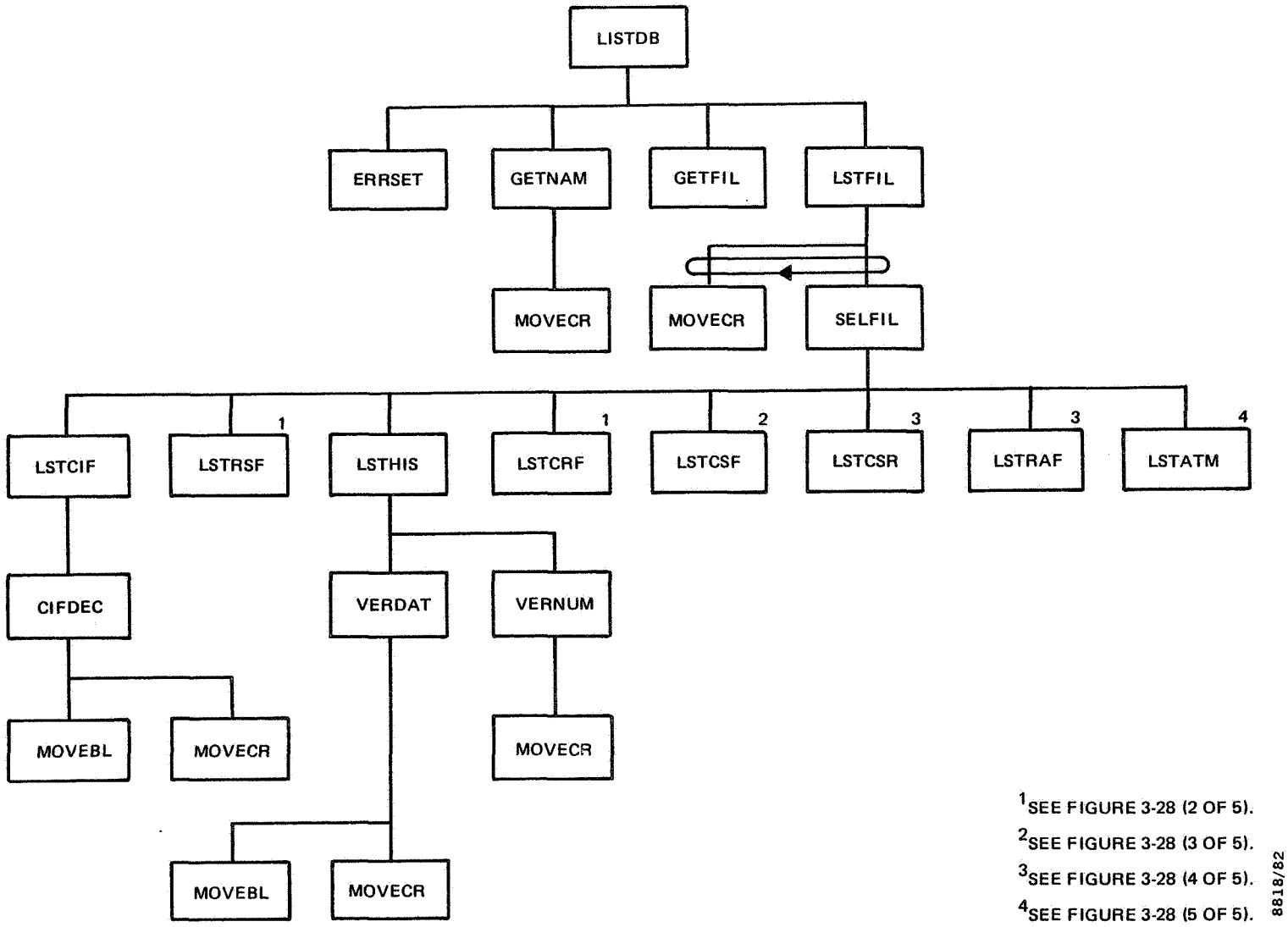


Figure 3-28. Baseline Diagram for the SEL Data Base Listing Program (LISTDB) (1 of 5)

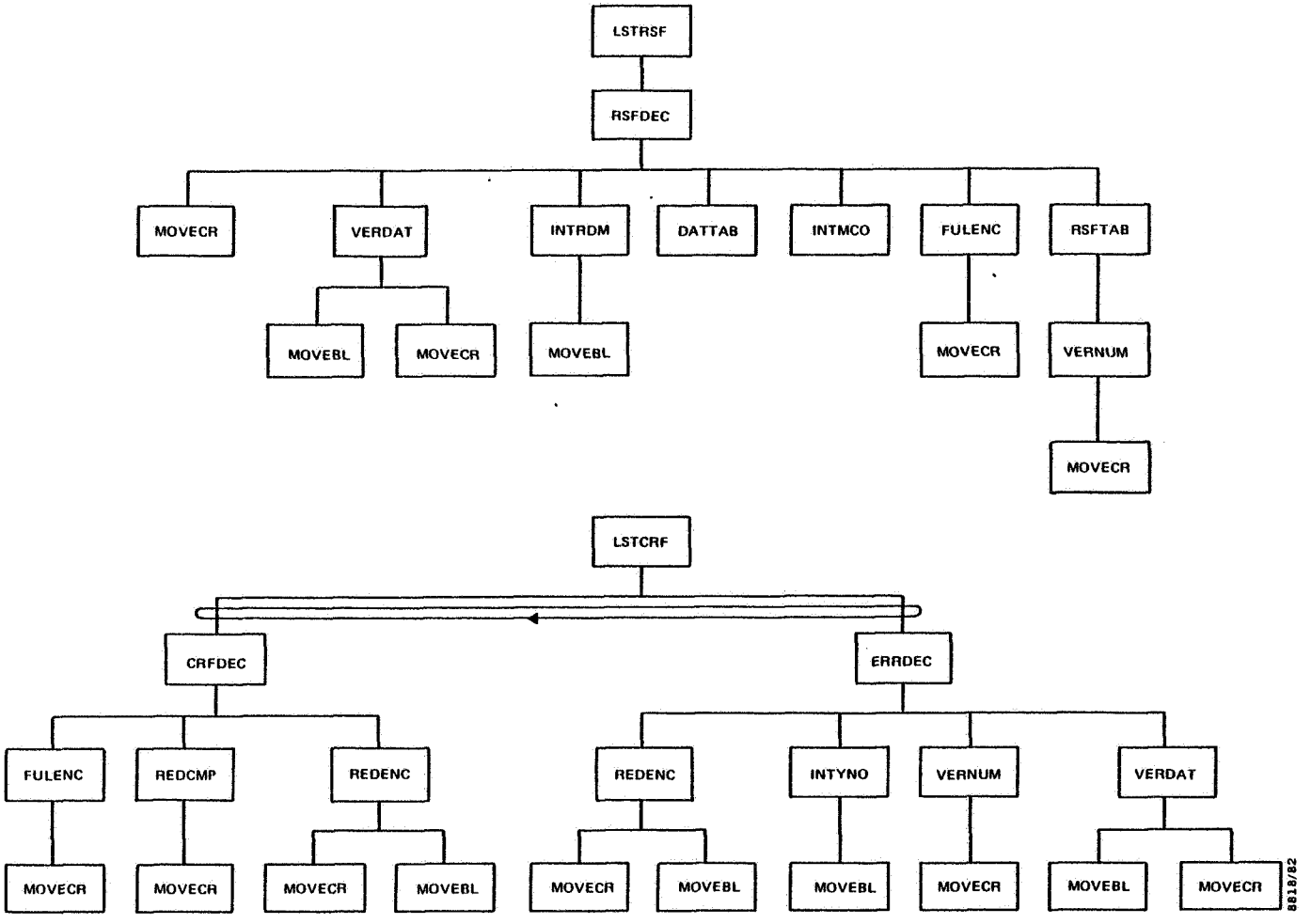


Figure 3-28. Baseline Diagram for the SEL Data Base Listing Program (LISTDB) (2 of 5)

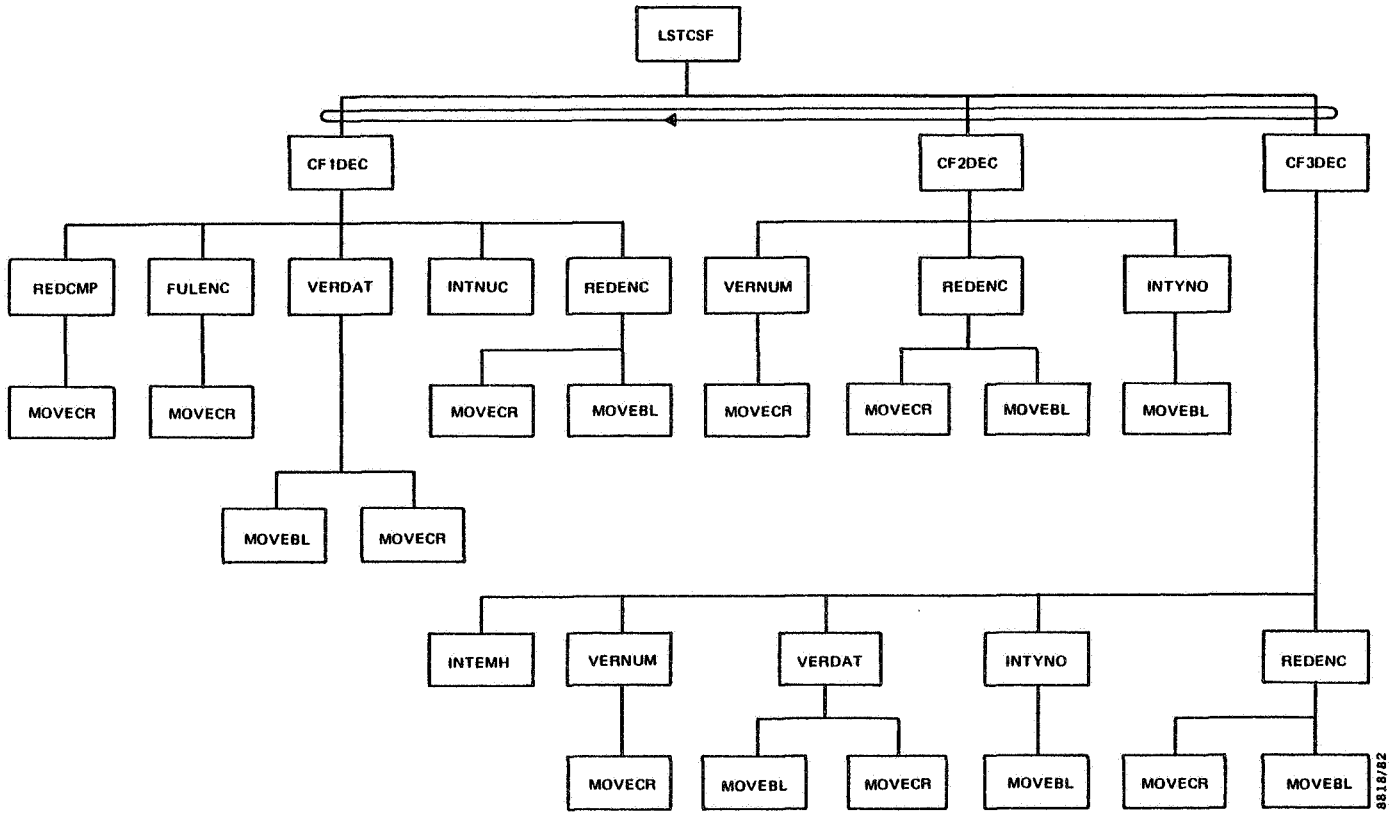


Figure 3-28. Baseline Diagram for the SEL Data Base Listing Program (LISTDB) (3 of 5)

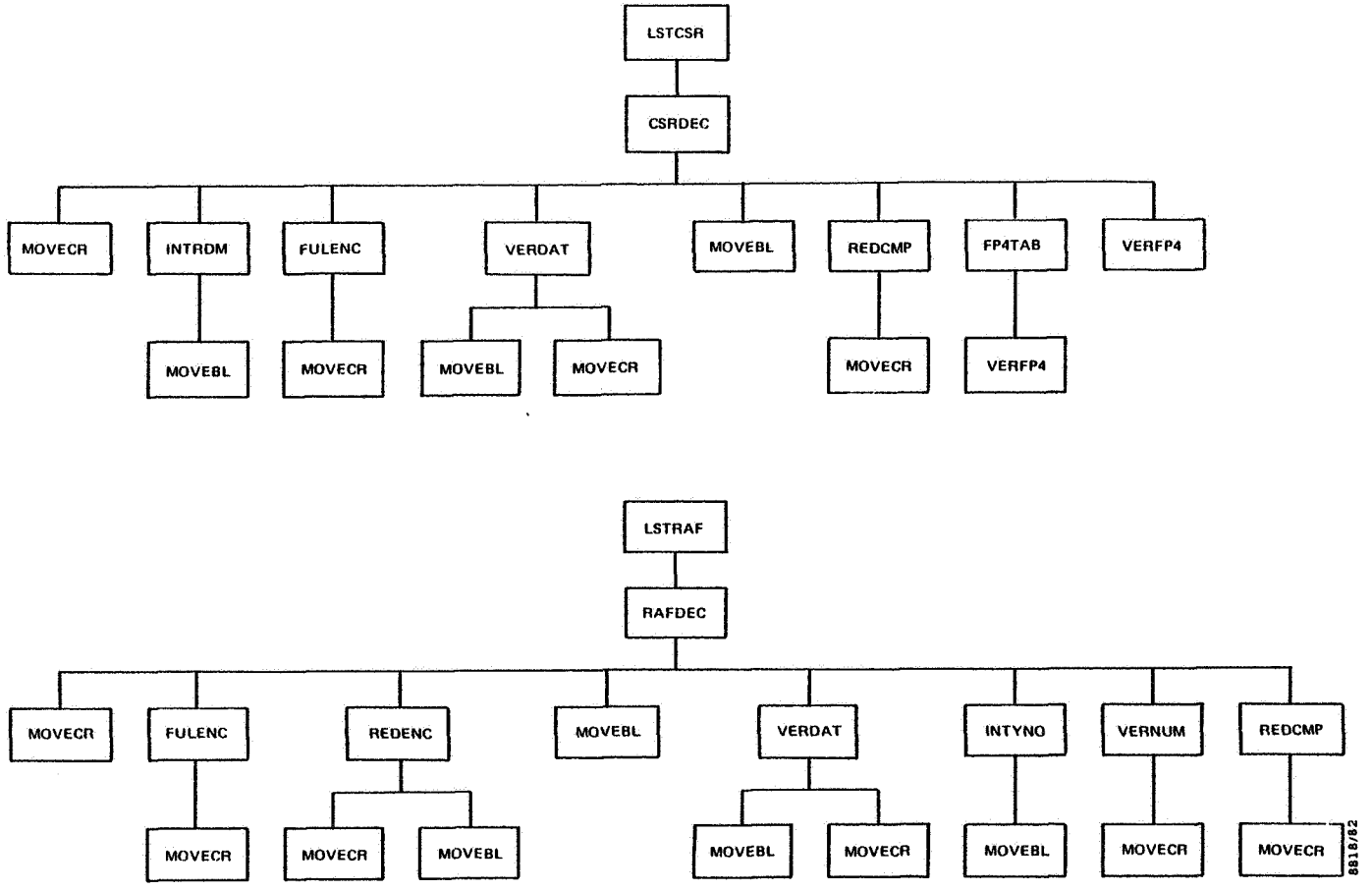


Figure 3-28. Baseline Diagram for the SEL Data Base Listing Program (LISTDB) (4 of 5)

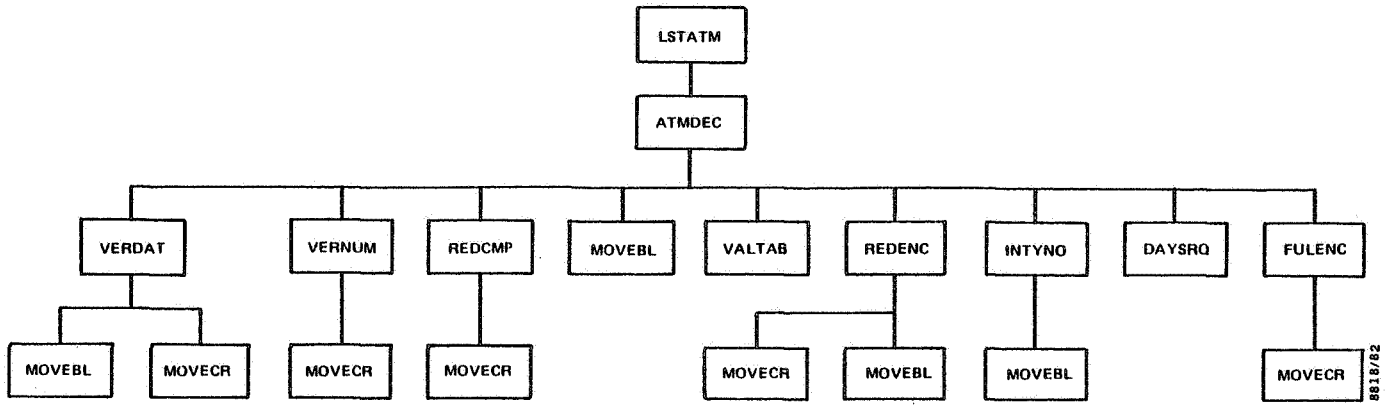


Figure 3-28. Baseline Diagram for the SEL Data Base Listing Program (LISTDB) (5 of 5)

ROUTINE: ATMDEC

FUNCTION: Decodes and verifies the fields of an ATM file record

CALLING SEQUENCE:

```
CALL ATMDEC (ATMREC, ENCREC, ENCKEY, PRTLIN, COMPS,
             LABELS, LUCIF, LUENC)
```

ROUTINE: CF1DEC

FUNCTION: Decodes part one of a CSF file record

CALLING SEQUENCE:

```
CALL CF1DEC (CSFREC, PRTLIN, ENCREC, ENCKEY, LABELS,
             COMPS, LUCIF, LUENC)
```

ROUTINE: CF2DEC

FUNCTION: Decodes part two of a CSF file record

CALLING SEQUENCE:

```
CALL CF2DEC (CSFREC, PRTLIN, ENCREC, ENCKEY, LABELS,
             LUENC)
```

ROUTINE: CF3DEC

FUNCTION: Decodes part three of a CSF file record

CALLING SEQUENCE:

```
CALL CF3DEC (CSFREC, PRTLIN, ENCREC, ENCKEY, LABELS,
             LUENC)
```

ROUTINE: CIFDEC

FUNCTION: Decodes and validates the fields of a CIF record

CALLING SEQUENCE:

```
CALL CIFDEC (CIFREC, ENCREC, ENCKEY, LABELS, LUENC)
```

ROUTINE: CRFDEC

FUNCTION: Decodes the primary fields of a CRF file record

CALLING SEQUENCE:

```
CALL CRFDEC (CRFREC, PRTLIN, ENCREC, ENCKEY, LABELS,
             PRNAME, LUCIF, LUENC)
```

ROUTINE: CSRDEC

FUNCTION: Decodes the fields of a CSR file record

CALLING SEQUENCE:

```
CALL CSRDEC (CSRREC, PRTLIN, ENCREC, ENCKEY, LFOR,
             LABELS, LUCIF, LUENC)
```

ROUTINE: ERRDEC

FUNCTION: Decodes the fields of the CRF error report

CALLING SEQUENCE:

```
CALL ERRDEC (CRFREC, PRTLIN, ENCREC, ENCKEY, LABELS,
             LUENC)
```

ROUTINE: LISTDB

FUNCTION: Main driver of the LISTDB program, produces formatted lists of SEL data base files

CALLING SEQUENCE: None

ROUTINE: LSTATM

FUNCTION: Reads, decodes, and displays records from the ATM file

CALLING SEQUENCE:

```
CALL LSTATM (PRNAME, ATMREC, ENCREC, ENCKEY, LUCIF,
             LUATM, LUENC, LUDSP)
```

ROUTINE: LSTCIF

FUNCTION: Reads, decodes, and displays CIF records

CALLING SEQUENCE:

```
CALL LSTCIF (PRNAME, CIFREC, ENCREC, ENCKEY, LUCIF,
             LUENC, LUDSP)
```

ROUTINE: LSTCRF

FUNCTION: Reads, decodes, and displays CRF file records and also displays an error report if indicated

CALLING SEQUENCE:

```
CALL LSTCRF (PRNAME, CRFREC, ENCREC, ENCKEY, LUCIF,
             LUCRF, LUENC, LUDSP, LUERR)
```

ROUTINE: LSTCSF

FUNCTION: Reads, decodes, and displays CSF file records in three parts

CALLING SEQUENCE:

```
CALL LSTCSF (PRNAME, CSFREC, ENCREC, ENCKEY, LUCIF,
             LUCSF, LUENC, LUDS1, LUDS2, LUDS3)
```

ROUTINE: LSTCSR

FUNCTION: Reads, decodes, and displays CSR file records

CALLING SEQUENCE:

```
CALL LSTCSR (PRNAME, CSRREC, ENCREC, ENCKEY, LUCIF,
             LUCSR, LUENC, LUDSP)
```

ROUTINE: LSTFIL

FUNCTION: Constructs file names and reads and prints file contents

CALLING SEQUENCE:

```
CALL LSTFIL (NAMTAB, FILIND, PROTAB, NPRO)
```

ROUTINE: LSTHIS

FUNCTION: Reads, decodes, and displays HIS file records

CALLING SEQUENCE:

```
CALL LSTHIS (PRNAME, HISREC, LUHIS, LUDSP)
```

ROUTINE: LSTRAF

FUNCTION: Reads, decodes, and displays RAF file records

CALLING SEQUENCE:

```
CALL LSTRAF (PRNAME, RAFREC, ENCREC, ENCKEY, LUCIF,
             LURAF, LUENC, LUDSP)
```

ROUTINE: LSTRSF

FUNCTION: Reads, decodes, and validates RSF file data

CALLING SEQUENCE:

```
CALL LSTRSF (PRNAME, RSFREC, ENCREC, ENCKEY, LURSF,
             LUENC, LUDSP)
```

ROUTINE: RAFDEC

FUNCTION: Decodes and verifies an RAF file record

CALLING SEQUENCE:

```
CALL RAFDEC (RAFREC, PRTLIN, ENCREC, ENCKEY, LFOR,
             LABELS, COMPS, LUCIF, LUENC)
```

ROUTINE: RSFDEC

FUNCTION: Decodes and displays an RSF file record

CALLING SEQUENCE:

```
CALL RSFDEC (RSFREC, ENCREC, ENCKEY, LFOR, PRNAME,
             LUDSP, LUENC)
```

ROUTINE: SELFIL

FUNCTION: Opens a data base file and calls the corresponding read/display routine

CALLING SEQUENCE:

```
CALL SELFIL (PRNAME, FLNAME, IT, LUDBS, LUENC,
             LUDSP, LUALT, LUOPT, LUCIF)
```

3.9.3.2 Decode or Verify Data

These 16 routines mainly decode or verify a data field.

ROUTINE: DATTAB

FUNCTION: Computes 10 dates at 7-day intervals subsequent to the start date

CALLING SEQUENCE:

```
CALL DATTAB (START, DATES)
```

ROUTINE: DAYSRQ

FUNCTION: Decodes time-to-implement field for the ATM file record

CALLING SEQUENCE:

```
CALL DAYSRQ (INBYT, OUTFLD)
```

ROUTINE: FP4TAB

FUNCTION: Decodes numeric fields for the CSR file record

CALLING SEQUENCE:

```
CALL FP4TAB (INFLD, OUTFLD, NFL)
```

ROUTINE: FULENC

FUNCTION: Converts numeric codes to alphabetic equivalents using the Encoding Dictionary

CALLING SEQUENCE:

```
CALL FULENC (LDATA, LTYPE, ENCREC, ENCKEY, LABELS, NVAL,
             LUENC)
```

ROUTINE: INTEMH

FUNCTION: Decodes the complexity field for the CSF file record

CALLING SEQUENCE:

```
CALL INTEMH (INBYT, OUTFLD)
```

ROUTINE: INTMCO

FUNCTION: Interprets resource type

CALLING SEQUENCE:

```
CALL INTMCO (INBYT, LSTBYT, OUTFLD, NUM)
```

ROUTINE: INTNUC

FUNCTION: Decodes the form stage field for the CSF file record

CALLING SEQUENCE:

```
CALL INTNUC (INBYT, OUTFLD)
```

ROUTINE: INTRDM

FUNCTION: Interprets phase flag

CALLING SEQUENCE:

```
CALL INTRDM (INFLD, OUTFLD)
```

ROUTINE: INTYNO

FUNCTION: Interprets yes-no responses

CALLING SEQUENCE:

```
CALL INTYNO (INBYT, OUTFLD)
```

ROUTINE: REDCMP

FUNCTION: Converts numeric codes to alphabetic equivalents using the CIF

CALLING SEQUENCE:

```
CALL REDCMP (LDATA, CIFREC, CIFKEY, LABELS, NVAL, LUCIF)
```

ROUTINE: REDENC

FUNCTION: Converts numeric codes to alphabetic equivalents using the Encoding Dictionary

CALLING SEQUENCE:

```
CALL REDENC (LDATA, LTYPE, ENCREC, ENCKEY, LABELS, NVAL,
             LUENC)
```

ROUTINE: RSFTAB

FUNCTION: Verifies resource fields

CALLING SEQUENCE:

```
CALL RSFTAB (RSFREC, HRSLIN, RUNLIN, RTYPE)
```

ROUTINE: VALTAB

FUNCTION: Decodes change types and error activities for the ATM file record

CALLING SEQUENCE:

```
CALL VALTAB (INFLD, OUTFLD, NAMTAB, NFL, TBYT)
```

ROUTINE: VERDAT

FUNCTION: Verifies date

CALLING SEQUENCE:

```
CALL VERDAT (INFLD, OUTFLD)
```

ROUTINE: VERFP4

FUNCTION: Verifies numeric field

CALLING SEQUENCE:

```
CALL VERFP4 (INFLD, OUTFLD)
```

ROUTINE: VERNUM

FUNCTION: Decodes a numeric field

CALLING SEQUENCE:

```
CALL VERNUM (INFLD, OUTFLD, FLEN)
```

3.9.3.3 Obtain Data From Terminal

These two routines obtain information from a user's response to a terminal prompt.

ROUTINE: GETFIL

FUNCTION: Prompts for, validates, and marks file names

CALLING SEQUENCE:

```
CALL GETFIL (NAMTAB, NFIL, FILIND)
```

ROUTINE: GETNAM

FUNCTION: Prompts for project names, checks them against the Encoding Dictionary, and saves them in a table

CALLING SEQUENCE:

```
CALL GETNAM (PROTAB, NPRO)
```

3.9.3.4 Routine With String Movement

These 2 routines deal with string movement.

ROUTINE: MOVEBL

FUNCTION: Moves blanks to an array of specified length

CALLING SEQUENCE:

```
CALL MOVEBL (VALUE, LENGTH)
```

ROUTINE: MOVECR

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVECR (INBUFF, OUTBUF, LENGTH)
```

3.9.3.5 Variable Description

The variables in the calling sequences of main LISTDB routines are described below.

Name	Type	Description
ATMREC(77)	L*1	Buffer array to hold an ATM file record
CIFKEY(3)	L*1	Tertiary key for the CIF (component code)
CIFREC(80)	L*1	Buffer array to hold a CIF record
COMPS(11)	R*8	Array containing component names
CRFREC(101)	L*1	Buffer array to hold a CRF file record
CSFREC(250)	L*1	Buffer array to hold a CSF file record
CSRREC(79)	L*1	Buffer array to hold a CSR file record
DATES(22)	I*2	Dates (M1, D1, M2, D2, ..., M11, D11)
ENCKEY(8)	L*1	Primary key for the Encoding Dictionary (code type and code)

Name	Type	Description
ENCREC(60)	L*1	Buffer array to hold an Encoding Dictionary record
FILIND(8)	L*1	Flag indicating whether a given file is to be listed or not
FLEN	I*2	Length of a given numeric field
FLNAME(23)	L*1	File name
HISREC(29)	L*1	Buffer array to hold an HIS file record
HRSLIN(58)	L*1	Array containing number of hours used for runs
INBYT	L*1	Input character
INFLD(X)	L*1	Input characters (length X is variable, dependent on the length of a particular field)
IT	I*2	File identification number = 1, CIF = 2, CRF = 3, CSF = 4, CSR = 5, RAF = 6, RSF = 7, HIS = 8, ATM
LABELS(X)	R*8	Decoded value for a field (length X is variable)
LDATA(X)	L*1	Input numeric codes that are to be converted to alphabetic names using the Encoding Dictionary or CIF (length X is variable)
LFOR(6)	L*1	Decoded form number
LSTBYT	L*1	Previous resource type
LTYPE(X)	L*1	Code type on Encoding Dictionary (length X is variable, X must be multiple of 3)
LUALT	I*2	Unit number for the second output listing file
LUATM	I*2	ATM file unit number
LUCIF	I*2	CIF unit number
LUCRF	I*2	CRF file unit number
LUCSF	I*2	CSF file unit number

Name	Type	Description
LUCSR	I*2	CSR file unit number
LUDBS	I*2	Unit number for a given data base file
LUDSP	I*2	Output report file unit number
LUDS1	I*2	CSF file output report part one unit number
LUDS2	I*2	CSF file output report part two unit number
LUDS3	I*2	CSF file output report part three unit number
LUENC	I*2	Unit number for Encoding Dictionary
LUERR	I*2	Unit number for the error report of the CRF file
LUHIS	I*2	Unit number for the HIS file
LUOPT	I*2	Unit number for the third output listing file
LURAF	I*2	RAF file unit number
LURSF	I*2	RSF file unit number
NAMTAB(8)	R*4	File name table
NFIL	I*2	Number of files to be listed
NFL	I*2	Number of fields
NPRO	I*2	Number of projects
NUM	I*2	Code type indicator for RSF record
NVAL	I*2	Number of bytes of a given field to be decoded
OUTFLD(X)	L*1	Decoded output characters (length X is variable)
PRNAME	R*8	Project name
PROTAB(20)	R*8	Project name array
PRTLIN(X)	L*1	Decoded output characters (length X is variable)
RAFREC(53)	L*1	Buffer array to hold an RAF file record
RSFREC(115)	L*1	Buffer array to hold an RSF file record
RTYPE	L*1	Resource type

Name	Type	Description
RUNLIN(33)	L*1	Decoded number of runs for computer resource
START(6)	L*1	First date
TBYT(2)	L*1	Error detection activities identifier D = detection, I = isolation, B = both

3.9.4 TASK BUILD PROCEDURE

3.9.4.1 Command Procedures

The LISTDB program can be generated from the source code by executing the command procedure [204,6]DLGEN.CMD. This command procedure references three command files--DLFPP.CMD, DLFOR.CMD, and LISTDB.TKB--all under UIC [204,6]. Figure 3-29 is a listing of DLGEN.CMD, the command procedure to precompile, compile, and task build the LISTDB program. The LISTDB program is generated by executing the following command:

```
@[204,6]DLGEN
```

3.9.4.2 Overlay Structure

The LISTDB program is overlaid to reduce the memory space requirement. Figure 3-30 is a listing of the Overlay Descriptor Language file, [204,6]LISTDB.ODL, needed to build the LISTDB program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;    @DLGEN.CMD                                                            2
;                                                                          3
;    COMMAND PROCEDURE TO GENERATE THE SEL DATA BASE LISTING PROGRAM       4
;    (LISTDB) FROM THE SOURCE CODES    (P. LO    7/21/82)                  5
;                                                                          6
;    PRECOMPILE FORTRAN ROUTINES                                           7
;                                                                          8
@[204,6]DLFPP.CMD                                                          9
;                                                                         10
;    @DLFPP.CMD                                                           11
;                                                                         12
;    COMMAND PROCEDURE TO PRECOMPILE ALL FORTRAN ROUTINES FOR THE SEL      13
;    DATA BASE LISTING PROGRAM (LISTDB)    (P. LO    7/21/82)            14
;                                                                         15
;    ROUTINE WITH PREFIX DL                                               16
;                                                                         17
;FPP SY:[204,6]DLATMDEC                                                   18
;FPP SY:[204,6]DLCF1DEC                                                   19
;FPP SY:[204,6]DLCF2DEC                                                   20
;FPP SY:[204,6]DLCF3DEC                                                   21
;FPP SY:[204,6]DLCIFDEC                                                   22
;FPP SY:[204,6]DLCRFDEC                                                   23
;FPP SY:[204,6]DLCSRDEC                                                   24
;FPP SY:[204,6]DLDATTAB                                                   25
;FPP SY:[204,6]DLDAYSRQ                                                   26
;FPP SY:[204,6]DLERRDEC                                                   27
;FPP SY:[204,6]DLFP4TAB                                                   28
;FPP SY:[204,6]DLFULENC                                                   29
;FPP SY:[204,6]DLGETFIL                                                   30
;FPP SY:[204,6]DLGETNAM                                                   31
;FPP SY:[204,6]DLINTEMH                                                   32
;FPP SY:[204,6]DLINTMCO                                                   33
;FPP SY:[204,6]DLINTNUC                                                   34
;FPP SY:[204,6]DLINTRDM                                                   35
;FPP SY:[204,6]DLINTYNO                                                   36
;FPP SY:[204,6]DLLISTDB                                                   37
;FPP SY:[204,6]DLLSTATM                                                   38
;FPP SY:[204,6]DLLSTCIF                                                   39
;FPP SY:[204,6]DLLSTCRF                                                   40
;FPP SY:[204,6]DLLSTCSF                                                   41
;FPP SY:[204,6]DLLSTCSR                                                   42
;FPP SY:[204,6]DLLSTFIL                                                   43
;FPP SY:[204,6]DLLSTHIS                                                   44
;FPP SY:[204,6]DLLSTRAF                                                   45
;FPP SY:[204,6]DLLSTRSF                                                   46
;FPP SY:[204,6]DLRAFDEC                                                   47
;FPP SY:[204,6]DLREDCMP                                                   48
;FPP SY:[204,6]DLREDENC                                                   49
;FPP SY:[204,6]DLRSFDEC                                                   50
;FPP SY:[204,6]DLRSFTAB                                                   51
;FPP SY:[204,6]DLSELFIL                                                   52
;FPP SY:[204,6]DLVALTAB                                                   53
;FPP SY:[204,6]DLVERDAT                                                   54
;FPP SY:[204,6]DLVERFP4                                                   55
```

Figure 3-29. LISTDB Task Generation Command Procedure (DLGEN.CMD) (1 of 3)

```
;FPP SY:[204,6]DLVERNUM                                                    56
;                                                                          57
;    ROUTINE WITH PREFIX DM                                                58
;                                                                          59
;FPP SY:[204,15]DMMOVEBL                                                   60
;                                                                          61
;    COMPILE FORTRAN ROUTINES                                              62
;                                                                          63
@[204,6]DLFOR.CMD                                                          64
;                                                                          65
;    @DLFOR.CMD                                                            66
;                                                                          67
;    COMMAND PROCEDURE TO COMPILE ALL FORTRAN ROUTINES FOR THE SEL DATA    68
;    BASE LISTING PROGRAM (LISTDB)    (P. LO    7/21/82)                   69
;                                                                          70
;    ROUTINE WITH PREFIX DL                                                71
;                                                                          72
;FOR/F4P/OBJECT:[204,6]DLATMDEC [204,6]DLATMDEC                            73
;FOR/F4P/OBJECT:[204,6]DLCF1DEC [204,6]DLCF1DEC                            74
;FOR/F4P/OBJECT:[204,6]DLCF2DEC [204,6]DLCF2DEC                            75
;FOR/F4P/OBJECT:[204,6]DLCF3DEC [204,6]DLCF3DEC                            76
;FOR/F4P/OBJECT:[204,6]DLCIFDEC [204,6]DLCIFDEC                            77
;FOR/F4P/OBJECT:[204,6]DLCRFDEC [204,6]DLCRFDEC                            78
;FOR/F4P/OBJECT:[204,6]DLCSRDEC [204,6]DLCSRDEC                            79
;FOR/F4P/OBJECT:[204,6]DLDATTAB [204,6]DLDATTAB                            80
;FOR/F4P/OBJECT:[204,6]DLDAYSRQ [204,6]DLDAYSRQ                            81
;FOR/F4P/OBJECT:[204,6]DLERRDEC [204,6]DLERRDEC                            82
;FOR/F4P/OBJECT:[204,6]DLFP4TAB [204,6]DLFP4TAB                            83
;FOR/F4P/OBJECT:[204,6]DLFULENC [204,6]DLFULENC                            84
;FOR/F4P/OBJECT:[204,6]DLGETFIL [204,6]DLGETFIL                            85
;FOR/F4P/OBJECT:[204,6]DLGETNAM [204,6]DLGETNAM                            86
;FOR/F4P/OBJECT:[204,6]DLINTEMH [204,6]DLINTEMH                            87
;FOR/F4P/OBJECT:[204,6]DLINTMCO [204,6]DLINTMCO                            88
;FOR/F4P/OBJECT:[204,6]DLINTNUC [204,6]DLINTNUC                            89
;FOR/F4P/OBJECT:[204,6]DLINTRDM [204,6]DLINTRDM                            90
;FOR/F4P/OBJECT:[204,6]DLINTYNO [204,6]DLINTYNO                            91
;FOR/F4P/OBJECT:[204,6]DLLISTDB [204,6]DLLISTDB                            92
;FOR/F4P/OBJECT:[204,6]DLLSTATM [204,6]DLLSTATM                            93
;FOR/F4P/OBJECT:[204,6]DLLSTCIF [204,6]DLLSTCIF                            94
;FOR/F4P/OBJECT:[204,6]DLLSTCRF [204,6]DLLSTCRF                            95
;FOR/F4P/OBJECT:[204,6]DLLSTCSF [204,6]DLLSTCSF                            96
;FOR/F4P/OBJECT:[204,6]DLLSTCSR [204,6]DLLSTCSR                            97
;FOR/F4P/OBJECT:[204,6]DLLSTFIL [204,6]DLLSTFIL                            98
;FOR/F4P/OBJECT:[204,6]DLLSTHIS [204,6]DLLSTHIS                            99
;FOR/F4P/OBJECT:[204,6]DLLSTRAF [204,6]DLLSTRAF                           100
;FOR/F4P/OBJECT:[204,6]DLLSTRSF [204,6]DLLSTRSF                           101
;FOR/F4P/OBJECT:[204,6]DLRAFDEC [204,6]DLRAFDEC                           102
;FOR/F4P/OBJECT:[204,6]DLREDCMP [204,6]DLREDCMP                           103
;FOR/F4P/OBJECT:[204,6]DLREDENC [204,6]DLREDENC                           104
;FOR/F4P/OBJECT:[204,6]DLRSFDEC [204,6]DLRSFDEC                           105
;FOR/F4P/OBJECT:[204,6]DLRSFTAB [204,6]DLRSFTAB                           106
;FOR/F4P/OBJECT:[204,6]DLSELFIL [204,6]DLSELFIL                           107
;FOR/F4P/OBJECT:[204,6]DLVALTAB [204,6]DLVALTAB                           108
;FOR/F4P/OBJECT:[204,6]DLVERDAT [204,6]DLVERDAT                           109
;FOR/F4P/OBJECT:[204,6]DLVERFP4 [204,6]DLVERFP4                           110
```

Figure 3-29. LISTDB Task Generation Command Procedure (DLGEN.CMD) (2 of 3)

```
;FOR/F4P/OBJECT:[204,6]DLVERNUM [204,6]DLVERNUM                          111
;                                                                       112
;   ROUTINE WITH PREFIX DM                                              113
;                                                                       114
;FOR/F4P/OBJECT:[204,15]DMMOVEBL [204,15]DMMOVEBL                       115
;                                                                       116
;    COMPILE ASSEMBLER ROUTINE                                          117
;                                                                       118
MAC/OBJECT:[204,7]UTCHAREQ [204,7]UTCHAREQ                              119
;                                                                       120
;   TASK BUILD THE LISTDB PROGRAM                                       121
;                                                                       122
TKB @[204,6]LISTDB.TKB                                                  123
;                                                                       124
;   @LISTDB.TKB                                                         125
;                                                                       126
;   COMMAND PROCEDURE TO TASK BUILD THE SEL DATA BASE LISTING PROGRAM    127
;   (LISTDB)                                                            128
;                                                                       129
;[204,5]LISTDB=[204,6]LISTDB/MP                                         130
;UNITS=8                                                                131
;ACTFIL=6                                                               132
;MAXBUF=250                                                             133
;//                                                                     134
```

Figure 3-29. LISTDB Task Generation Command Procedure (DLGEN.CMD) (3 of 3)

```
;                                                                              1
;    @LISTDB.ODL                                                               2
;                                                                              3
;    OVERLAY DESCRIPTOR LANGUAGE FOR THE SEL DATA BASE LISTING PROGRAM         4
;    (LISTDB)                                                                  5
;                                                                              6
       .ROOT $R1,OTSALL,RMSALL                                                 7
$R1:   .FCTR [204,6]DLLISTDB-RMSROT-OTSROT-$R2                                 8
$R2:   .FCTR [204,7]UTCHAREQ-[204,15]DMMOVEBL-*($PO,$P1)                       9
$PO:   .FCTR [204,6]DLGETNAM-[204,6]DLGETFIL                                   10
$P1:   .FCTR [204,6]DLLSTFIL-$P2-*($L1,$L2,$L3,$L4,$L5,$L6,$L7)               11
$P2:   .FCTR [204,6]DLSELFIL-[204,6]DLREDENC-[204,6]DLFULENC-$P3               12
$P3:   .FCTR [204,6]DLREDCMP-[204,6]DLVERDAT-[204,6]DLVERNUM-$P4               13
$P4:   .FCTR [204,6]DLFP4TAB-[204,6]DLVERFP4-[204,6]DLINTYNO-$P5               14
$P5:   .FCTR [204,6]DLINTRDM                                                   15
$L1:   .FCTR [204,6]DLLSTCIF-[204,6]DLCIFDEC                                   16
$L2:   .FCTR [204,6]DLLSTCRF-[204,6]DLCRFDEC-[204,6]DLERRDEC                   17
$L3:   .FCTR [204,6]DLLSTCSF-*($L31,[204,6]DLCF2DEC,$L32)                      18
$L31:  .FCTR [204,6]DLCF1DEC-[204,6]DLINTNUC                                   19
$L32:  .FCTR [204,6]DLCF3DEC-[204,6]DLINTEMH                                   20
$L4:   .FCTR [204,6]DLLSTCSR-[204,6]DLCSRDEC-[204,6]DLLSTHIS                   21
$L5:   .FCTR [204,6]DLLSTRAF-[204,6]DLRAFDEC                                   22
$L6:   .FCTR [204,6]DLLSTRSF-[204,6]DLRSFDEC-[204,6]DLDATTAB-$L61              23
$L61:  .FCTR [204,6]DLINTMCO-[204,6]DLRSFTAB                                   24
$L7:   .FCTR [204,6]DLLSTATM-[204,6]DLATMDEC-[204,6]DLVALTAB-$L71              25
$L71:  .FCTR [204,6]DLDAYSRQ                                                   26
;                                                                              27
@LB:[1,1]RMS11M.ODL                                                            28
@LB:[1,1]RMS12X.ODL                                                            29
       .END                                                                    30
```

Figure 3-30. LISTDB Program Overlay Descriptor Language File (LISTDB.ODL)

3.10 SEL DATA BASE RECENT ACTIVITY REPORT PROGRAM (RC)

3.10.1 INTRODUCTION

The SEL Data Base Recent Activity Report Program (RC) generates a one-page report of the additions, deletions and changes to records in the SEL data base since the last back-up date. This information is retrieved from the transaction files.

3.10.2 PROGRAM STRUCTURE

3.10.2.1 Files Accessed

The RC program accesses eight input files and one output file as described below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file
DB0:[204,1]TRANS.CIF	Component Information Transaction file
DB0:[204,1]TRANS.CRF	Change Report Form Transaction file
DB0:[204,1]TRANS.CSF	Component Summary Form Transaction file
DB0:[204,1]TRANS.CSR	Component Status Report Transaction file
DB0:[204,1]TRANS.HIS	Growth History Transaction file
DB0:[204,1]TRANS.RAF	Run Analysis Form Transaction file
DB0:[204,1]TRANS.RSF	Resource Summary Form Transaction file

Output File Name	Description
RECENT.RPT	Recent activity output report file

3.10.2.2 Baseline Diagram

Figure 3-31 is the baseline diagram for the RC program. The RECENT routine is the main driver. It obtains the project name from the Encoding Dictionary; counts all adds, deletes, and changes from the transaction files for a given project; and then generates a report of all adds, deletes, and changes for all projects.

3.10.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the RC program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any), and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major RC routines are described in Section 3.10.3.6. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the RC program also uses the following system routines: DATE, ERRSET, SECNDS, and TIME.

3.10.3.1 Process Data and Compute Statistics

These two major routines count all adds, deletes, and changes in the transaction files for all projects.

ROUTINE: RECENT

FUNCTION: Main routine of the RC program, generates a one-page report of the additions, deletions, and changes to records in the SEL data base

CALLING SEQUENCE: None

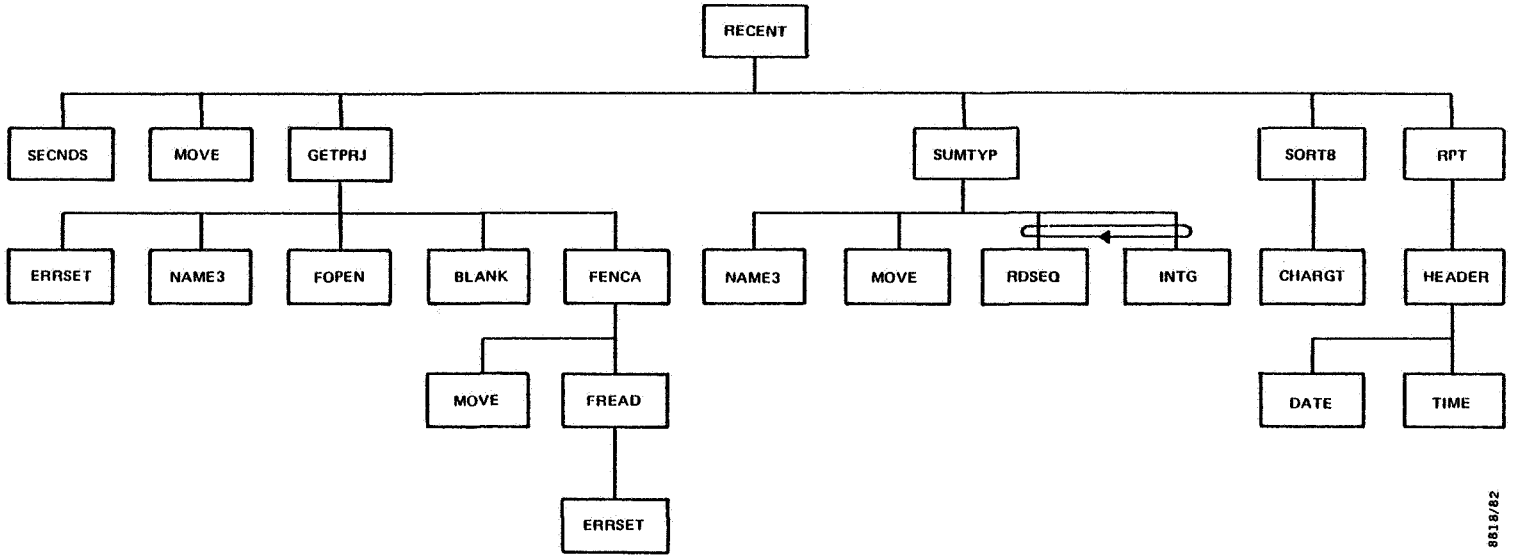


Figure 3-31. Baseline Diagram for the SEL Data Base Recent Activity Report Program (RC)

ROUTINE: SUMTYP

FUNCTION: Obtains a count of all additions, deletions, and changes to the given file type in the data base from the transaction files

CALLING SEQUENCE:

```
CALL SUMTYP (ITYP, LOC, MAXACT, MAXPRJ, MAXTYP, RECL, TYP,
             COUNTS,
             DATE)
```

3.10.3.2 Write Output Report

These two routines write a one-page report of the additions, deletions, and changes for all projects.

ROUTINE: HEADER

FUNCTION: Prints a one-line title for each report page that includes the date and project

CALLING SEQUENCE:

```
CALL HEADER (IRPTF, PRJNAM, RPTITL)
```

ROUTINE: RPT

FUNCTION: Prints a one-page report of the transaction file counts

CALLING SEQUENCE:

```
CALL RPT (COUNTS, DATE, MAXACT, MAXPRJ, MAXTYP, PRJNAM,
          SRTKEY)
```

3.10.3.3 File Open and Read Routines

These five routines either open an indexed file or read records from an indexed file.

ROUTINE: FENCA

FUNCTION: Finds the description field on the Encoding Dictionary corresponding to the given type and code

CALLING SEQUENCE:

```
CALL FENCA (IENCF, TYPE, CODE,
            NAME, REST, FOUND)
```

ROUTINE: FOPEN

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL FOPEN (IUNIT, FILNAM,
            ERROR)
```

ROUTINE: FREAD

FUNCTION: Reads one indexed record

CALLING SEQUENCE:

```
CALL FREAD (IUNIT, KEYVAL, KEYLEN, LRECL,
            BUFFER, ERROR)
```

ROUTINE: GETPRJ

FUNCTION: Obtains all project names from the Encoding Dictionary

CALLING SEQUENCE:

```
CALL GETPRJ (MAXPRJ,
             PRJNAM)
```

ROUTINE: RDSEQ

FUNCTION: Reads one record from a sequential file

CALLING SEQUENCE:

```
CALL RDSEQ (IUNIT, NCHARS,
            CHARS, EOF)
```

3.10.3.4 Sort Routine

This one routine provides a sort function.

ROUTINE: SORT8

FUNCTION: Generates an array of indices to alphabetize the given name array

CALLING SEQUENCE:

```
CALL SORT8 (MAX, NSORT, NAMES,
            SRTKEY)
```

3.10.3.5 Routines for String Movement or Comparison

These five routines deal with string movement or comparison.

ROUTINE: BLANK

FUNCTION: Initializes an array to blanks

CALLING SEQUENCE:

```
CALL BLANK (ARRAY, NUM)
```

ROUTINE: CHARGT (LOGICAL FUNCTION)

FUNCTION: Determines whether the first string alphabetically follows the second

CALLING SEQUENCE:

```
CHARGT (STRNG1, STRNG2, LEN)
```

ROUTINE: INTG (INTEGER*2 FUNCTION)

FUNCTION: Converts the given characters to integer

CALLING SEQUENCE:

```
INTG (BUFFER, LEN)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

ROUTINE: NAME3

FUNCTION: Concatenates the given strings to form a complete file name

CALLING SEQUENCE:

```
CALL NAME3 (DISK, UIC, NAME, EXTENS,
            DSN)
```

3.10.3.6 Variable Description

The variables in the calling sequences of main RC routines are described below.

Name	Type	Description
COUNTS (MAXACT, MAXTYP, MAXPRJ)	I*2	Count of all additions, deletions, and changes of all data base files as recorded on the transaction files
DATE(6)	L*1	Last backup date
ITYP	I*2	Number of current file type
LOC	I*2	Location of the field of the project code within a record
MAX	I*2	Maximum number of project names
MAXACT	I*2	Total number of activities (add, delete, change, total)
MAXPRJ	I*2	Maximum number of projects
MAXTYP	I*2	Total number of files + 1
NAMES(8,NSORT)	L*1	Names to be sorted
NSORT	I*2	Number of names to be sorted

Name	Type	Description
PRJNAM(8,MAXPRJ)	L*1	Project names
RECL	I*2	Logical record length for a given transaction file
SRTKEY(MAXPRJ)	I*2	Sort index array to alphabetize project names
TYP(3)	L*1	Current file type (e.g., 'CIF')

3.10.4 TASK BUILD PROCEDURE

3.10.4.1 Command Procedure

The RC program can be generated from the source code by executing the command procedure RCGEN.CMD under UIC [204,6]. This command procedure references three command procedures--RCFPP.CMD, RCFOR.CMD, and RC.TKB--all under UIC [204,6]. Figure 3-32 is a listing of RCGEN.CMD, the command procedure to precompile, compile, and task build the RC program. The RC program is generated by executing the following command:

```
@[204,6]RCGEN
```

3.10.4.2 Overlay Structure

The RC program is overlaid to reduce the memory space requirement. Figure 3-33 is a listing of the Overlay Descriptor Language file, [204,6]RC.ODL, needed to build the RC program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @RCGEN.CMD                                                             2
;                                                                          3
;   COMMAND PROCEDURE TO GENERATE THE RECENT ACTIVITY REPORT PROGRAM        4
;   (RC) FROM THE SOURCE CODES    (P. LO    7/30/82)                       5
;                                                                          6
;   PRECOMPILE ALL STRUCTURED FORTRAN SOURCE CODES                         7
;                                                                          8
@[204,6]RCFPP                                                              9
;                                                                         10
;   @RCFPP.CMD                                                            11
;                                                                         12
;   COMMAND PROCEDURE TO PRECOMPILE ALL ROUTINES WRITTEN IN STRUCTURED     13
;   FORTRAN FOR THE SEL DATA BASE RECENT ACTIVITY REPORT PROGRAM (RC)      14
;   (P. LO    7/30/82)                                                    15
;                                                                         16
;   ROUTINES WITH PREFIX RC                                               17
;                                                                         18
;FPP SY:[204,6]RCGETPRJ                                                   19
;FPP SY:[204,6]RCRECENT                                                   20
;FPP SY:[204,6]RCRPT                                                      21
;FPP SY:[204,6]RCSORT8                                                    22
;FPP SY:[204,6]RCSUMTYP                                                   23
;                                                                         24
;   ROUTINES WITH PREFIX UT                                               25
;                                                                         26
;FPP SY:[204,7]UTBLANK                                                    27
;FPP SY:[204,7]UTCHARGT                                                   28
;FPP SY:[204,7]UTFENCA                                                    29
;FPP SY:[204,7]UTFOPEN                                                    30
;FPP SY:[204,7]UTFREAD                                                    31
;FPP SY:[204,7]UTHEADER                                                   32
;FPP SY:[204,7]UTINTG                                                     33
;FPP SY:[204,7]UTMOVE                                                     34
;FPP SY:[204,7]UTNAME3                                                    35
;FPP SY:[204,7]UTRDSEQ                                                    36
;                                                                         37
;   COMPILE ALL FORTRAN ROUTINES                                          38
;                                                                         39
@[204,6]RCFOR                                                             40
;                                                                         41
;   @RCFOR.CMD                                                            42
;                                                                         43
;   COMMAND PROCEDURE TO COMPILE ALL FORTRAN ROUTINES FOR THE SEL DATA     44
;   BASE RECENT ACTIVITY REPORT PROGRAM (RC)                              45
;   (P. LO    7/30/82)                                                    46
;                                                                         47
;   ROUTINES WITH PREFIX RC                                               48
;                                                                         49
;FOR/F4P/OBJECT:[204,6]RCGETPRJ [204,6]RCGETPRJ                           50
;FOR/F4P/OBJECT:[204,6]RCRECENT [204,6]RCRECENT                           51
;FOR/F4P/OBJECT:[204,6]RCRPT    [204,6]RCRPT                              52
;FOR/F4P/OBJECT:[204,6]RCSORT8  [204,6]RCSORT8                            53
;FOR/F4P/OBJECT:[204,6]RCSUMTYP [204,6]RCSUMTYP                           54
;                                                                         55
```

Figure 3-32. RC Task Generation Command Procedure (RCGEN.CMD) (1 of 2)

```
;   ROUTINES WITH PREFIX UT                                                    56
;                                                                              57
;FOR/F4P/OBJECT:[204,7]UTBLANK   [204,7]UTBLANK                                 58
;FOR/F4P/OBJECT:[204,7]UTCHARGT  [204,7]UTCHARGT                                59
;FOR/F4P/OBJECT:[204,7]UTFENCA   [204,7]UTFENCA                                 60
;FOR/F4P/OBJECT:[204,7]UTFOPEN   [204,7]UTFOPEN                                 61
;FOR/F4P/OBJECT:[204,7]UTFREAD   [204,7]UTFREAD                                 62
;FOR/F4P/OBJECT:[204,7]UTHEADER  [204,7]UTHEADER                                63
;FOR/F4P/OBJECT:[204,7]UTINTG    [204,7]UTINTG                                  64
;FOR/F4P/OBJECT:[204,7]UTMOVE    [204,7]UTMOVE                                  65
;FOR/F4P/OBJECT:[204,7]UTNAME3   [204,7]UTNAME3                                 66
;FOR/F4P/OBJECT:[204,7]UTRDSEQ   [204,7]UTRDSEQ                                 67
;                                                                              68
;   TASK BUILD THE RC PROGRAM                                                  69
;                                                                              70
TKB @[204,6]RC.TKB                                                             71
;                                                                              72
;   @RC.TKB                                                                    73
;                                                                              74
;   COMMAND PROCEDURE TO TASK BUILD THE RECENT ACTIVITY REPORT PROGRAM          75
;   (RC)                                                                       76
;                                                                              77
;[204,5]RC=[204,6]RC/MP                                                        78
;MAXBUF=263                                                                    79
://                                                                            80
```

Figure 3-32. RC Task Generation Command Procedure (RCGEN.CMD) (2 of 2)

```
;                                                                          1
;   @RC.ODL                                                                2
;                                                                          3
;   THE OVERLAY DESCRIPTOR LANGUAGE FOR THE RECENT ACTIVITY REPORT         4
;   PROGRAM (RC)                                                           5
;                                                                          6
        .ROOT RMSROT-OTSROT-$ROOT,OTSALL,RMSALL                            7
$ROOT:  .FCTR [204, 6]RCGETPRJ-[204, 6]RCRECENT-[204, 6]RCRPT   -$ROOT2    8
$ROOT2: .FCTR [204, 6]RCSORT8 -[204, 6]RCSUMTYP-[204, 7]UTINTG  -$ROOT4    9
$ROOT4: .FCTR [204, 7]UTBLANK -[204, 7]UTMOVE  -$ROOT5                     10
$ROOT5: .FCTR [204, 7]UTNAME3 -[204, 7]UTHEADER-[204, 7]UTFOPEN -$ROOT8    11
$ROOT8: .FCTR [204, 7]UTFENCA -[204, 7]UTFREAD -$ROOT9                     12
$ROOT9: .FCTR [204, 7]UTRDSEQ -[204, 7]UTCHARGT                            13
;                                                                          14
@LB:[1,1]RMS11M                                                            15
@LB:[1,1]RMS12X                                                            16
;                                                                          17
        .END                                                               18
```

Figure 3-33. RC Program Overlay Descriptor Language File (RC.ODL)

3.11 SEL DATA BASE RECORD COUNTING REPORT PROGRAM (RPSTSCTR)

3.11.1 INTRODUCTION

The SEL Data Base Record Counting Program (RPSTSCTR) counts the number of records in each file in the SEL data base and produces a one-page report of all counts.

3.11.2 PROGRAM STRUCTURE

3.11.2.1 Files Accessed

The RPSTSCTR program accesses all SEL data base files as input files and produces one output report file. In addition, the user's copy of the File Name and Status (STS) file is accessed as both an input and an output file.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file
[204,1]HEADER.HDR	Phase Dates (HDR) file
[204,1]STAT.HDR	STS file
[204,1]EST.HDR	Estimated Statistics (EST) file
[204,1]<PRJNAM>.CIF	Component Information File (CIF) for each project
[204,1]<PRJNAM>.CMT	Comment file for each project
[204,1]<PRJNAM>.CRF	Change Report Form (CRF) file for each project
[204,1]<PRJNAM>.CSF	Component Summary Form (CSF) file for each project
[204,1]<PRJNAM>.CSR	Component Status Report (CSR) file for each project
[204,1]<PRJNAM>.HIS	Growth History (HIS) file for each project
[204,1]<PRJNAM>.RAF	Run Analysis Form (RAF) file for each project
[204,1]<PRJNAM>.RSF	Resource Summary Form (RSF) file for each project
[User's UIC]STAT.HDR	User's copy of the STS file

In these file names, <PRJNAM> is the project name.

Output File Name	Description
STSCTR.RPT	Output report file
[User's UIC]STAT.HDR	User's copy of the STS file

3.11.2.2 Baseline Diagram

Figure 3-34 is the baseline diagram for the RPSTSCTR program. The STSCTR routine is the driver that opens all input files, counts the number of records in each file, and then writes the output report. It also updates the user's copy of the STS file.

3.11.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The RPSTSCTR program references only two routines, STSCTR (main routine) and MOVECR, in addition to two system routines (DATE, SECNDS) and the RMSIAC routines. These two routines are described below. However, descriptions of the calling sequence variables for MOVECR are not provided.

ROUTINE: STSCTR

FUNCTION: Main routine of the RPSTSCTR program, counts the number of records in each file in the SEL data base and produces a one-page report of all counts

CALLING SEQUENCES: NONE

ROUTINE: MOVECR

FUNCTION: Moves given number of characters from one address to another.

CALLING SEQUENCES:

```
CALL MOVECR (INBUFF, OTBUFF, LENGTH)
```

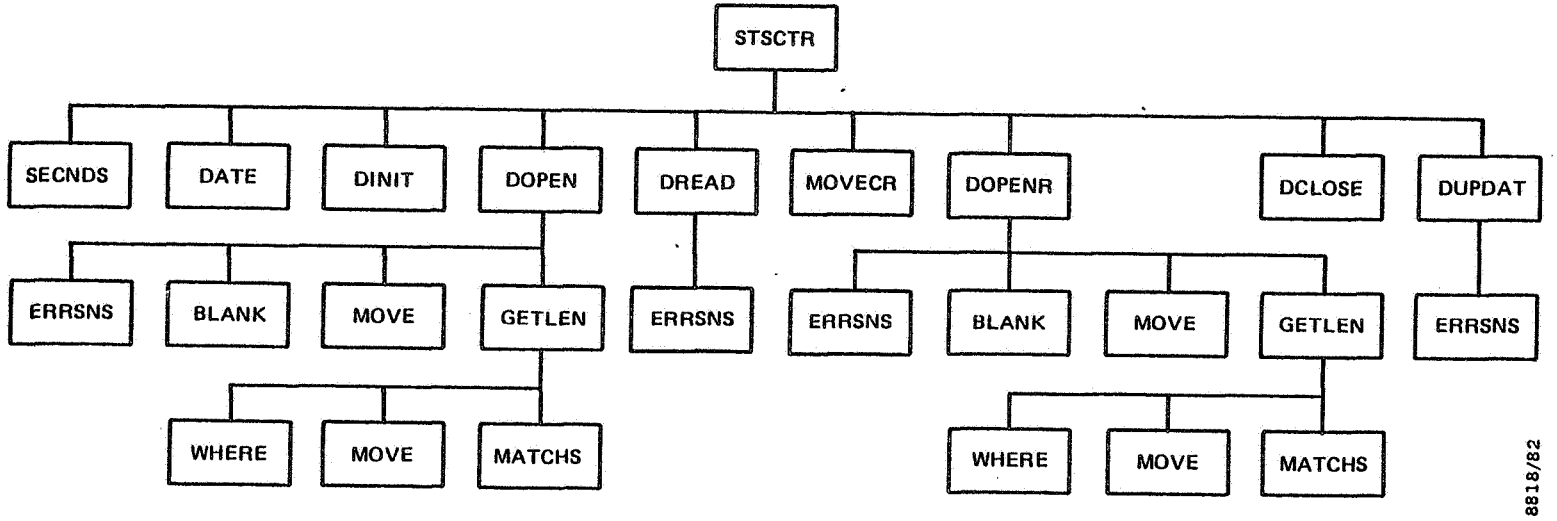


Figure 3-34. Baseline Diagram for the SEL Data Base Record Counting Report Program (RPSTSCTR)

3.11.4 TASK BUILD PROCEDURE

3.11.4.1 Command Procedures

The RPSTSCTR program can be generated from the source code by executing the command procedure RPSTSGEN.CMD under UIC [204,6]. Figure 3-35 is a listing of this command procedure, which precompiles and compiles the FORTRAN routine, compiles the ASSEMBLER routine, and task builds the RPSTSCTR program. RPSTSGEN.CMD references another command procedure, RPSTSCTR.TKB, also under UIC [204,6], which builds the RPSTSCTR program task image. The RPSTSCTR program is generated by entering the following command:

```
@[204,6]RPSTSGEN
```

3.11.4.2 Overlay Structure

The RPSTSCTR task is overlaid to reduce the memory space requirement. Figure 3-36 is a listing of the Overlay Descriptor Language file, [204,6]RPSTSCTR.ODL, needed to build the RPSTSCTR program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay. In addition, the RMS Indexed Access Programs Library (RMSIAC), [204,7]UFRMSIAC.OLB, is also needed in the overlay. This library contains FORTRAN routines used to access RMS indexed files.

```
;                                                                          1
;   @RPSTSGEN.CMD                                                          2
;                                                                          3
;   COMMAND PROCEDURE TO GENERATE THE SEL DATA BASE RECORD COUNTING        4
;   PROGRAM (RPSTSCTR) FROM SOURCE CODES                                   5
;   (P. LO     8/11/82)                                                    6
;                                                                          7
;   PRECOMPILE STRUCTURED FORTRAN SOURCE CODES                             8
;                                                                          9
FPP SY:[204,6]RPSTSCTR                                                    10
;                                                                         11
;   COMPILE FORTRAN SOURCE CODES                                          12
;                                                                         13
FOR/F4P/OBJECT:[204,6]RPTSTCTP [204,6]RPSTSCTR                            14
;                                                                         15
;   COMPILE ASSEMBLER ROUTINE                                             16
;                                                                         17
MAC/OBJECT:[204,7]UTCHAREQ [204,7]UTCHAREQ                                18
;                                                                         19
;   TASK BUILD THE RPSTSCTR PROGRAM                                       20
;                                                                         21
TKB @[204,6]RPSTSCTR.TKB                                                  22
;                                                                         23
;   @RPSTSCTR.TKB                                                         24
;                                                                         25
;   COMMAND PROCEDURE TO BUILD THE SEL DATA BASE RECORD COUNTING          26
;   PROGRAM (RPSTSCTR)                                                    27
;                                                                         28
;[204,5]RPSTSCTR/FU,RPSTSCTR/NOSP/SH=[204,6]RPSTSCTR.ODL/MP                29
;ACTFIL=2                                                                 30
;UNITS=20                                                                 31
;MAXBUF=250                                                               32
;//                                                                       33
```

Figure 3-35. RPSTSCTR Task Generation Command Procedure (RPSTSGEN.CMD)

```
;                                                                          1
;   @RPSTSCTR.ODL                                                          2
;                                                                          3
;   THE OVERLAY STRUCTURE FOR THE SEL DATA BASE RECORD COUNTING            4
;   PROGRAM (RPSTSCTR)                                                     5
;   (P. LO     8/11/82)                                                    6
;                                                                          7
        .ROOT RMSROT-OTSROT-$ROOT,OTSALL,RMSALL                            8
$ROOT:  .FCTR [204,6]RPSTSCTR-[204,7]UTCHAREQ-[204,7]UFRMSIAC/LB           9
;                                                                         10
;                                                                         11
@LB:[1,1]RMS11M.ODL                                                       12
@LB:[1,1]RMS12X.ODL                                                       13
        .END                                                              14
```

Figure 3-36. RPSTSCTR Program Overlay Descriptor Language File (RPSTSCTR.ODL)

3.12 COMPONENT NAME REPORT GENERATOR PROGRAM (RPCOMPNM)

3.12.1 INTRODUCTION

The Component Name Report Generator Program (RPCOMPNM) reads all Component Information Files (CIFs) on the SEL data base and produces a formatted and alphabetized report of component names and codes for all such files.

3.12.2 PROGRAM STRUCTURE

3.12.2.1 Files Accessed

The RPCOMPNM program accesses all CIFs and the Encoding Dictionary as the input files and one output file.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file
[204,1]<PRJNAM>.CIF	CIF for each project, where <PRJNAM> is the project name

Output File Name	Description
COMPNAMES.RPT	Output report file

3.12.2.2 Baseline Diagram

Figure 3-37 is the baseline diagram for the RPCOMPNM program. The COMRPT routine is the driver that opens all input files, reads the desired data from the files, and writes the output report.

3.12.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

In addition to two system routines (DATE, SECNDS) and the RMSIAC routines, the RPCOMPNM program references only one routine, the driver (COMRPT), as described below.

ROUTINE: COMRPT

FUNCTION: Reads component names and codes from CIFs and writes a formatted report of all components for all projects.

CALLING SEQUENCES: None

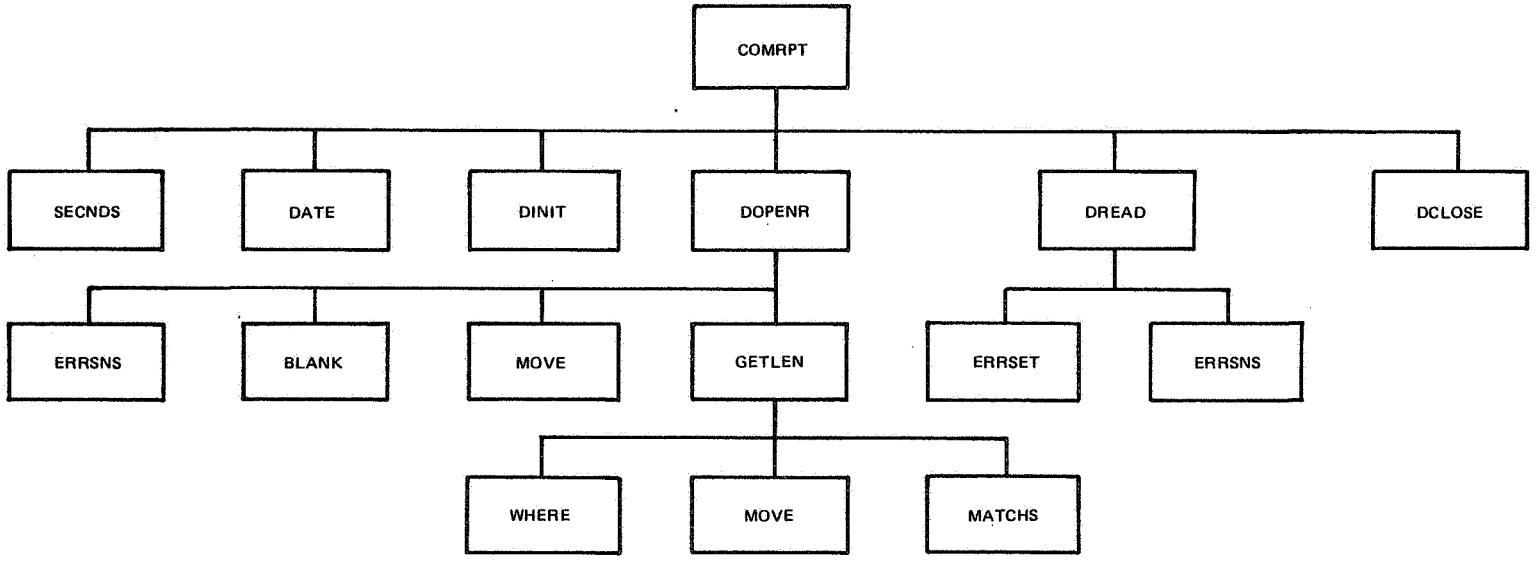


Figure 3-37. Baseline Diagram for the Component Name Report Generator Program (RPCOMPNM)

3.12.4 TASK BUILD PROCEDURE

3.12.4.1 Command Procedures

The RPCOMPNM program can be generated from the source code by executing the command procedure RPCOMGEN.CMD under UIC [204,6]. This command procedure precompiles and compiles the FORTRAN routines and task builds the RPCOMPNM program. It references another command procedure, RPCOMPNM.TKB, also under UIC [204,6], which builds the RPCOMPNM program task image. Figure 3-38 is a listing of RPCOMGEN.CMD. The RPCOMPNM program is generated by executing the following command:

```
@[204,6]RPCOMGEN
```

3.12.4.2 Overlay Structure

The RPCOMPNM program is overlaid to reduce the memory space requirement. Figure 3-39 is a listing of the Overlay Descriptor Language file, [204,6]RPCOMPNM.ODL, needed to build the RPCOMPNM program task image. The system libraries RMS11M.ODL and RMS12X.ODL and the RMS Indexed Access Programs Library (RMSIAC) are needed in the overlay. The name of this last library is UFRMSIAC.OLB, under UIC [204,7]; it contains FORTRAN routines used to access RMS indexed files.

```
;                                                                          1
;   @RPCOMGEN.CMD                                                          2
;                                                                          3
;   COMMAND PROCEDURE TO BUILD THE COMPONENT NAME GENERATOR (RPCOMPNM)      4
;   TASK IMAGE FROM SOURCE CODE                                            5
;   (P. LO     9/9/82)                                                     6
;                                                                          7
;   PRECOMPILE FORTRAN ROUTINE                                             8
;                                                                          9
FPP SY:[204,6]RPCOMPNM                                                    10
;                                                                         11
;   COMPILE FORTRAN ROUTINE                                               12
;                                                                         13
FOR/F4P/OBJECT:[204,6]PRCOMPNM [204,6]RPCOMPNM                            14
;                                                                         15
;   TASK BUILD THE RPCOMPNM PROGRAM                                       16
;                                                                         17
TKB @[204,6]RPCOMPNM.TKB                                                  18
;                                                                         19
;   @RPCOMPNM.TKB                                                         20
;                                                                         21
;   COMMAND PROCEDURE TO BUILD THE TASK IMAGE FOR THE COMPONENT NAME      22
;   GENERATOR (RPCOMPNM)                                                  23
;                                                                         24
;[204,5]RPCOMPNM/FU=[204,6]RPCOMPNM/MP                                     25
;UNITS=20                                                                  26
;ACTFIL=2                                                                  27
;//                                                                       28
```

Figure 3-38. RPCOMPNM Task Generation Command Procedure (RPCOMGEN.CMD)

```
;                                                                          1
;   @RPCOMPNM.ODL                                                          2
;                                                                          3
;   THE OVERLAY STRUCTURE FOR THE COMPONENT NAME GENERATOR (RPCOMPNM)       4
;   (P. LO     9/9/82)                                                     5
;                                                                          6
        .ROOT $ROOT-RMSROT-OTSROT,OTSALL,RMSALL                            7
$ROOT:  .FCTR [204,6]RPCOMPNM-[204,7]UFRMSIAC/LB                           8
;                                                                          9
;                                                                         10
@LB:[1,1]RMS11M.ODL                                                       11
@LB:[1,1]RMS12X.ODL                                                       12
        .END                                                              13
```

Figure 3-39. RPCOMPNM Program Overlay Descriptor Language File (RPCOMPNM.ODL)

3.13 SUBJECTIVE EVALUATIONS FILE LISTING PROGRAM (DBRPTSEF)

3.13.1 INTRODUCTION

The Subjective Evaluations File Listing Program (DBRPTSEF) reads the Subjective Evaluations File (SEF) on the SEL data base and generates a formatted report of the contents of the file organized by the category of measure.

3.13.2 PROGRAM STRUCTURE

3.13.2.1 Files Accessed

The DBRPTSEF program accesses two input files and one output file as described below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary File
[204,1]SEF.HDR	Subjective Evaluations File

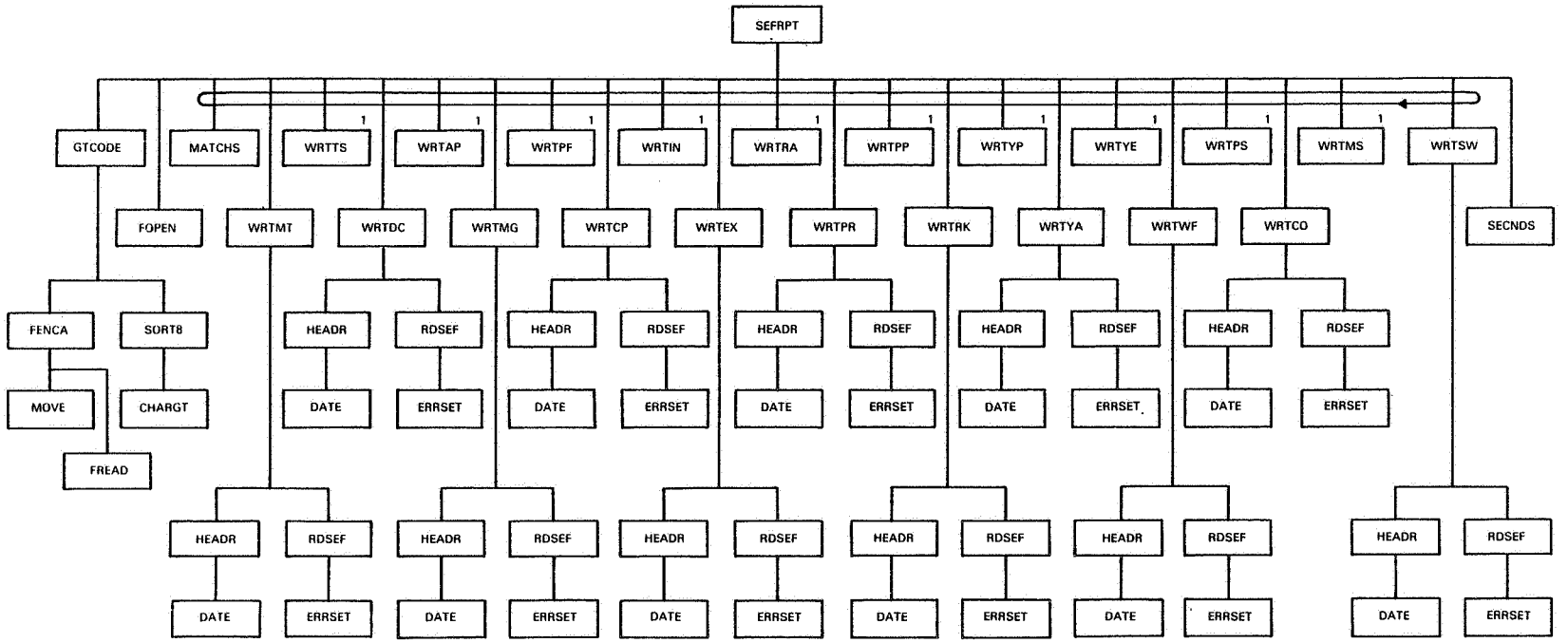
Output File Name	Description
[204,3]SEFDAT.RPT	Output listing of the contents of the SEF

3.13.2.2 Baseline Diagram

Figure 3-40 is the baseline diagram for the DBRPTSEF program. The SEFRPT routine is the main driver. It opens all files, obtains all project codes from the SEF and the corresponding project names from the Encoding Dictionary, obtains the user option for the category of measure to be listed, and then writes the selected listing from the SEF. It loops through this process until a^Z (control Z) is entered by the user in response to a prompt.

3.13.3 SUBROUTINE/SUBSYSTEM DESCRIPTION

The routines forming the DBRPTSEF program are grouped here by function. In each routine, the calling sequence variables are grouped according to input, input and output (if any),



SEE FIGURE 3-40 (2 OF 2)

Figure 3-40. Baseline Diagram for the Subjective Evaluations File Listing Program (DBRPTSEF) (1 of 2)

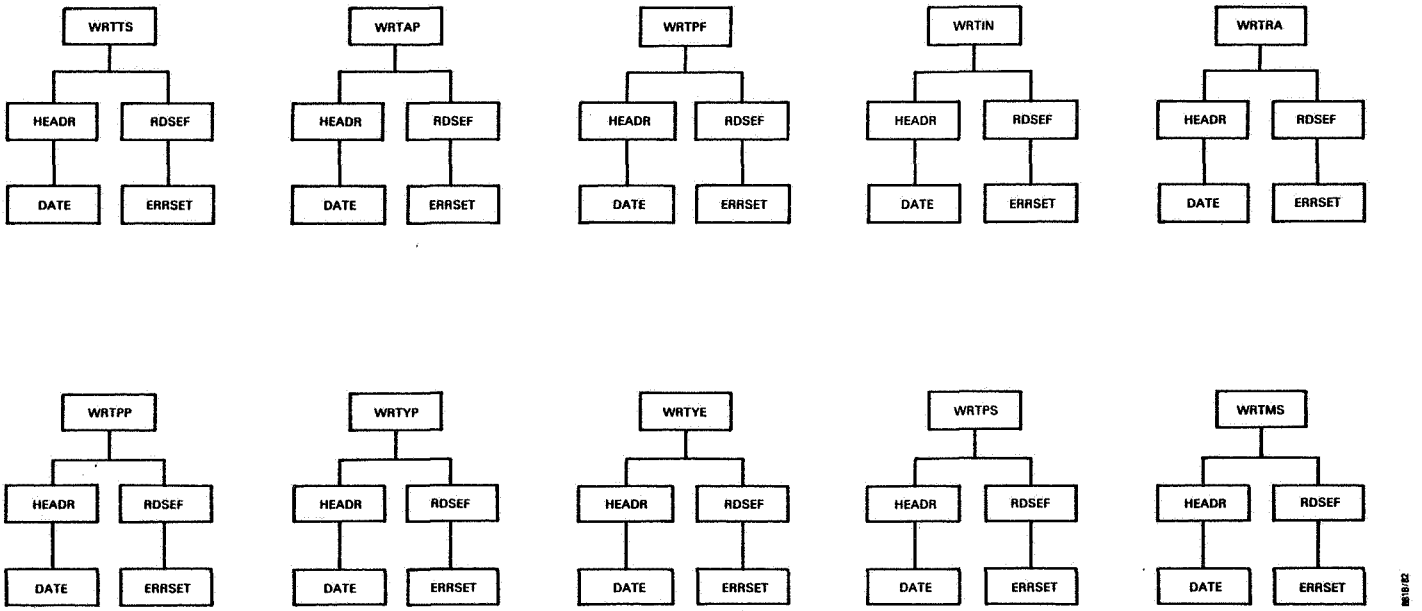


Figure 3-40. Baseline Diagram for the Subjective Evaluations File Listing Program (DBRPTSEF) (2 of 2)

and output and appear in the calling sequence in that order. In the following descriptions, each group of variables begins a new line. The calling sequence variables for the major DBRPTSEF routines are described in Section 3.13.3.5. Descriptions of the calling sequence variables for utility routines are not provided. In addition to the routines described in this section, the DBRPTSEF program also uses the following system routines: DATE, ERRSET, and SECNDS.

3.13.3.1 Process Data and Produce Formatted Listing

These 23 major routines process the SEF data and produce a formatted listing of the contents of the SEF.

ROUTINE: GTCODE

FUNCTION: Obtains all project codes from the SEF and the corresponding project names from the Encoding Dictionary and sorts them alphabetically

CALLING SEQUENCE:

```
CALL GTCODE (IENC, ISEF,
             PRCO, PROJ, IREC, ERROR)
```

ROUTINE: SEFRPT

FUNCTION: Main routine of the DBRPTSEF program, produces a formatted listing of the contents of the SEF organized by category of measure

CALLING SEQUENCE: None

ROUTINE: WRTAP

FUNCTION: Generates the output listing for the experience with application (AP) measure

CALLING SEQUENCE:

```
CALL WRTAP (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTCO

FUNCTION: Generates the output listing for the COCOMO (CO) model measure

CALLING SEQUENCE:

```
CALL WRTCO (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTCP

FUNCTION: Generates the output listing for the complexity of problem (CP) measure

CALLING SEQUENCE:

```
CALL WRTCP (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTDC

FUNCTION: Generates the output listing for the documentation (DC) measure

CALLING SEQUENCE:

```
CALL WRTDC (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTEX

FUNCTION: Generates the output listing for the external influences on project (EX) measure

CALLING SEQUENCE:

```
CALL WRTEX (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTIN

FUNCTION: Generates the output listing for the internal influences on project (IN) measure

CALLING SEQUENCE:

```
CALL WRTIN (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTMG

FUNCTION: Generates the output listing for the effectiveness of management (MG) measure

CALLING SEQUENCE:

```
CALL WRTMG (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTMS

FUNCTION: Generates the output listing for the miscellaneous (MS) measure

CALLING SEQUENCE:

```
CALL WRTMS (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTMT

FUNCTION: Generates the output listing for the practices and techniques (MT) measure

CALLING SEQUENCE:

```
CALL WRTMT (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTPF

FUNCTION: Generates the output listing for the performance of team (PF) measure

CALLING SEQUENCE:

```
CALL WRTPF (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTPP

FUNCTION: Generates the output listing for the product/ process performance (PP) measure

CALLING SEQUENCE:

```
CALL WRTPP (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTPR

FUNCTION: Generates the output listing for the software product (PR) measure

CALLING SEQUENCE:

```
CALL WRTPR (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTPS

FUNCTION: Generates the output listing for the PRICE S3 (PS) model measure

CALLING SEQUENCE:

```
CALL WRTPS (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTRA

FUNCTION: Generates the output listing for the resources available (RA) measure

CALLING SEQUENCE:

```
CALL WRTRA (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTRK

FUNCTION: Generates the output listing for the team rank (RK) measure

CALLING SEQUENCE:

```
CALL WRTRK (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTSW

FUNCTION: Generates the output listing for the code breakdown (SW) measure

CALLING SEQUENCE:

```
CALL WRTSW (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTTS

FUNCTION: Generates the output listing for the tools (TS) measure

CALLING SEQUENCE:

```
CALL WRTTS (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTWF

FUNCTION: Generates the output listing for the Walston-Felix (WF) model measure

CALLING SEQUENCE:

```
CALL WRTWF (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTYA

FUNCTION: Generates the output listing for the years of applicable experience (YA) measure

CALLING SEQUENCE:

```
CALL WRTYA (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTYE

FUNCTION: Generates the output listing for the years of environment experience (YE) measure

CALLING SEQUENCE:

```
CALL WRTYE (ISEF, IRPT, PROJ, PRCO, IREC)
```

ROUTINE: WRTYP

FUNCTION: Generates the output listing for the years of professional experience (YP) measure

CALLING SEQUENCE:

```
CALL WRTYP (ISEF, IRPT, PROJ, PRCO, IREC)
```

3.13.3.2 Input and Output Routines

These five routines perform either input or output functions.

ROUTINE: FENCA

FUNCTION: Finds the description field on the Encoding Dictionary for the given type and code

CALLING SEQUENCE:

```
CALL FENCA (IENCF, TYPE, CODE,
            NAME, REST, FOUND)
```

ROUTINE: FOPEN

FUNCTION: Opens an indexed file

CALLING SEQUENCE:

```
CALL FOPEN (IUNIT, FILNAM,
            ERROR)
```

ROUTINE: FREAD

FUNCTION: Reads one indexed record

CALLING SEQUENCE:

```
CALL FREAD (IUNIT, KEYVAL, KEYLEN, LRECL,
            BUFFER, ERROR)
```

ROUTINE: HEADR

FUNCTION: Prints a two-line title for each report page, including the date and page number

CALLING SEQUENCE:

```
CALL HEADR (IRPT, TITLE1, TITLE2,
            IPAGE)
```

ROUTINE: RDSEF

FUNCTION: Reads one record from the SEF

CALLING SEQUENCE:

```
CALL RDSEF (ISEF, KVAL,
            ERROR, BUF, LRECL)
```

3.13.3.3 Sort Routine

This routine provides a sort function.

ROUTINE: SORT8

FUNCTION: Generates an array of indices to alphabetize the given name array

CALLING SEQUENCE:

```
CALL SORT8 (MAX, NSORT, NAMES,
            SRTKEY)
```

3.13.3.4 Routines Performing String Movement or Comparison

These three routines deal with string movement or comparison.

ROUTINE: CHARGT (LOGICAL*1 FUNCTION)

FUNCTION: Determines if the first string is alphabetically after the second

CALLING SEQUENCE:

```
CHARGT (STRNG1, STRNG2, LEN)
```

ROUTINE: MATCHS (LOGICAL*1 FUNCTION)

FUNCTION: Determines whether two input strings match

CALLING SEQUENCE:

```
MATCHS (ARRAY1, ARRAY2, NBYTES)
```

ROUTINE: MOVE

FUNCTION: Moves a given number of bytes from one address to another

CALLING SEQUENCE:

```
CALL MOVE (A, B, LEN)
```

3.13.3.5 Variable Description

The variables in the calling sequences of major DBRPTSEF routines are described below

Name	Type	Description
BUF(578)	L*1	The SEF record buffer
ERROR	L*1	Error flag
IENC	I*2	FORTRAN unit number for the Encoding Dictionary
IPAGE	I*2	Page number
IREC	I*2	Number of projects
IRPT	I*2	FORTRAN unit number for the output report file
ISEF	I*2	FORTRAN unit number for the SEF
KVAL(3)	L*1	Key value
PRCO(70)	I*2	Array of project codes
PROJ(70)	R*8	Array of project names
TITLE1(40)	L*1	First title line for each report page
TITLE2(50)	L*1	Second title line for each report page

3.13.4 TASK BUILD PROCEDURE

3.13.4.1 Command Procedures

The DBRPTSEF program can be generated from the source code by executing the command procedure DBSEFGEN.CMD under UIC [204,6]. This command procedure references three command procedures--DBSEFFPP.CMD, DBSEFFOR.CMD, and DBRPTSEF.TKB--all under UIC [204,6]. Figure 3-41 is a listing of DBSEFGEN.CMD, the command procedure to precompile, compile, and task build the DBRPTSEF program. The DBRPTSEF task is generated by executing the following command:

```
@[204,6]DBSEFGEN
```

3.13.4.2 Overlay Structure

The DBRPTSEF program is overlaid to reduce the memory space requirement. Figure 3-42 is a listing of the Overlay Descriptor Language file, [204,6]DBRPTSEF.ODL, needed to build the DBRPTSEF program task image. The system libraries RMS11M.ODL and RMS12X.ODL are needed for the overlay.

```
;                                                                          1
;   @DBSEFGEN.CMD                                                          2
;                                                                          3
;   COMMAND PROCEDURE TO GENERATE THE SUBJECTIVE EVALUATIONS FILE          4
;   LISTING PROGRAM (DBRPTSEF) TASK IMAGE FROM SOURCE CODE                 5
;   (P. LO     9/9/82)                                                     6
;                                                                          7
;   PRECOMPILE FORTRAN ROUTINES                                            8
;                                                                          9
@[204,6]DBSEFFPP.CMD                                                      10
;                                                                         11
;   @DBSEFFPP.CMD                                                         12
;                                                                         13
;   COMMAND PROCEDURE TO PRECOMPILE ALL ROUTINES WRITTEN IN STRUCTURED    14
;   FORTRAN FOR THE SUBJECTIVE EVALUATIONS FILE LISTING PROGRAM           15
;   (DBRPTSEF)     (P. LO    9/9/82)                                      16
;                                                                         17
;   ROUTINE WITH PREFIX SF                                                18
;                                                                         19
;FPP SY:[204, 6]SFGTCODE                                                  20
;FPP SY:[204, 6]SFHEADR                                                   21
;FPP SY:[204, 6]SFRDSEF                                                   22
;FPP SY:[204, 6]SFSEFRPT                                                  23
;FPP SY:[204, 6]SFWRTAP                                                   24
;FPP SY:[204, 6]SFWRTCO                                                   25
;FPP SY:[204, 6]SFWRTCP                                                   26
;FPP SY:[204, 6]SFWRTDC                                                   27
;FPP SY:[204, 6]SFWRTEX                                                   28
;FPP SY:[204, 6]SFWRTIN                                                   29
;FPP SY:[204, 6]SFWRTMG                                                   30
;FPP SY:[204, 6]SFWRTMS                                                   31
;FPP SY:[204, 6]SFWRTMT                                                   32
;FPP SY:[204, 6]SFWRTPF                                                   33
;FPP SY:[204, 6]SFWRTPP                                                   34
;FPP SY:[204, 6]SFWRTPR                                                   35
;FPP SY:[204, 6]SFWRTPS                                                   36
;FPP SY:[204, 6]SFWRTRA                                                   37
;FPP SY:[204, 6]SFWRTRK                                                   38
;FPP SY:[204, 6]SFWRTSW                                                   39
;FPP SY:[204, 6]SFWRTTS                                                   40
;FPP SY:[204, 6]SFWRTWF                                                   41
;FPP SY:[204, 6]SFWRTYA                                                   42
;FPP SY:[204, 6]SFWRTYE                                                   43
;FPP SY:[204, 6]SFWRTYP                                                   44
;                                                                         45
;   ROUTINE WITH PREFIX DM, RC, OR UT                                     46
;                                                                         47
;FPP SY:[204, 6]RCSORT8                                                   48
;FPP SY:[204, 7]UTCHARGT                                                  49
;FPP SY:[204, 7]UTFENCA                                                   50
;FPP SY:[204, 7]UTFOPEN                                                   51
;FPP SY:[204, 7]UTFREAD                                                   52
;FPP SY:[204, 7]UTMATCHS                                                  53
;FPP SY:[204, 7]UTMOVE                                                    54
;FPP SY:[204,15]DMZFILL                                                   55
```

Figure 3-41. DBRPTSEF Task Generation Command Procedure (DBSEFGEN.CMD) (1 of 2)

```
;
;    COMPILE FORTRAN ROUTINES                                                    56
;                                                                                57
@[204,6]DBSEFFOR.CMD                                                            58
;                                                                                59
;    @DBSEFFOR.CMD                                                               60
;                                                                                61
;    COMMAND PROCEDURE TO COMPILE ALL FORTRAN ROUTINES FOR THE SUBJECTIVE        62
;    EVALUATIONS FILE LISTING PROGRAM (DBRPTSEF)                                 63
;    (P. LO    9/9/82)                                                           64
;                                                                                65
;    ROUTINE WITH PREFIX SF                                                      66
;                                                                                67
;FOR/F4P/OBJECT:[204,6]SFGTCODE  [204,6]SFGTCODE                                 68
;FOR/F4P/OBJECT:[204,6]SFHEADR   [204,6]SFHEADR                                  69
;FOR/F4P/OBJECT:[204,6]SFRDSEF   [204,6]SFRDSEF                                  70
;FOR/F4P/OBJECT:[204,6]SFSEFRPT  [204,6]SFSEFRPT                                 71
;FOR/F4P/OBJECT:[204,6]SFWRTAP   [204,6]SFWRTAP                                  72
;FOR/F4P/OBJECT:[204,6]SFWRTCO   [204,6]SFWRTCO                                  73
;FOR/F4P/OBJECT:[204,6]SFWRTCP   [204,6]SFWRTCP                                  74
;FOR/F4P/OBJECT:[204,6]SFWRTDC   [204,6]SFWRTDC                                  75
;FOR/F4P/OBJECT:[204,6]SFWRTEX   [204,6]SFWRTEX                                  76
;FOR/F4P/OBJECT:[204,6]SFWRTIN   [204,6]SFWRTIN                                  77
;FOR/F4P/OBJECT:[204,6]SFWRTMG   [204,6]SFWRTMG                                  78
;FOR/F4P/OBJECT:[204,6]SFWRTMS   [204,6]SFWRTMS                                  79
;FOR/F4P/OBJECT:[204,6]SFWRTMT   [204,6]SFWRTMT                                  80
;FOR/F4P/OBJECT:[204,6]SFWRTPF   [204,6]SFWRTPF                                  81
;FOR/F4P/OBJECT:[204,6]SFWRTPP   [204,6]SFWRTPP                                  82
;FOR/F4P/OBJECT:[204,6]SFWRTPR   [204,6]SFWRTPR                                  83
;FOR/F4P/OBJECT:[204,6]SFWRTPS   [204,6]SFWRTPS                                  84
;FOR/F4P/OBJECT:[204,6]SFWRTRA   [204,6]SFWRTRA                                  85
;FOR/F4P/OBJECT:[204,6]SFWRTRK   [204,6]SFWRTRK                                  86
;FOR/F4P/OBJECT:[204,6]SFWRTSW   [204,6]SFWRTSW                                  87
;FOR/F4P/OBJECT:[204,6]SFWRTTS   [204,6]SFWRTTS                                  88
;FOR/F4P/OBJECT:[204,6]SFWRTWF   [204,6]SFWRTWF                                  89
;FOR/F4P/OBJECT:[204,6]SFWRTYA   [204,6]SFWRTYA                                  90
;FOR/F4P/OBJECT:[204,6]SFWRTYE   [204,6]SFWRTYE                                  91
;FOR/F4P/OBJECT:[204,6]SFWRTYP   [204,6]SFWRTYP                                  92
;                                                                                93
;    ROUTINE WITH PREFIX DM, RC, OR UT                                           94
;                                                                                95
;FOR/F4P/OBJECT:[204,6]RCSORT8   [204,6]RCSORT8                                  96
;FOR/F4P/OBJECT:[204,7]UTCHARGT  [204,7]UTCHARGT                                 97
;FOR/F4P/OBJECT:[204,7]UTFENCA   [204,7]UTFENCA                                  98
;FOR/F4P/OBJECT:[204,7]UTFOPEN   [204,7]UTFOPEN                                  99
;FOR/F4P/OBJECT:[204,7]UTFREAD   [204,7]UTFREAD                                 100
;FOR/F4P/OBJECT:[204,7]UTMATCHS  [204,7]UTMATCHS                                101
;FOR/F4P/OBJECT:[204,7]UTMOVE    [204,7]UTMOVE                                  102
;FOR/F4P/OBJECT:[204,15]DMZFILL  [204,15]DMZFILL                                103
;                                                                               104
;    TASK BUILD THE DBRPTSEF PROGRAM                                            105
;                                                                               106
TKB @[204,6]DBRPTSEF.TKB                                                        107
;                                                                               108
;    @DBRPTSEF.TKB                                                              109
;                                                                               110
;    COMMAND PROCEDURE TO BUILD THE SUBJECTIVE EVALUATIONS FILE LISTING         111
;    PROGRAM (DBRPTSEF) TASK IMAGE                                              112
;    (P. LO    9/9/82)                                                          113
;                                                                               114
;[204,5]DBRPTSEF=[204,6]DBRPTSEF.ODL/MP                                         115
;MAXBUF=578                                                                     116
;//                                                                            117
                                                                                118
```

Figure 3-41. DBRPTSEF Task Generation Command Procedure (DBSEFGEN.CMD) (2 of 2)

```
;                                                                          1
;   @DBRPTSEF.ODL                                                          2
;                                                                          3
;   THE OVERLAY STRUCTURE FOR THE SUBJECTIVE EVALUATIONS FILE LISTING       4
;   PROGRAM (DBRPTSEF)                                                     5
;   (P. LO    9/9/82)                                                      6
;                                                                          7
          .ROOT RMSROT-OTSROT-$ROOT,OTSALL,RMSALL                          8
$ROOT:    .FCTR [204,6]SFSEFRPT-[204,7]UTFOPEN -[204,7]UTMATCHS-R1         9
R1:       .FCTR [204,6]SFHEADR -[204,6]SFRDSEF -[204,15]DMZFILL-R2        10
R2:       .FCTR *(CODE,WRT)                                               11
;                                                                         12
CODE:     .FCTR [204,6]SFGTCODE-*(READ,SORT)                              13
READ:     .FCTR [204,7]UTFENCA -[204,7]UTMOVE  -[204,7]UTFREAD            14
SORT:     .FCTR [204,6]RCSORT8 -[204,7]UTCHARGT                           15
;                                                                         16
WRT:      .FCTR *(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U)              17
A:        .FCTR [204,6]SFWRTAP                                            18
B:        .FCTR [204,6]SFWRTCO                                            19
C:        .FCTR [204,6]SFWRTCP                                            20
D:        .FCTR [204,6]SFWRTDC                                            21
E:        .FCTR [204,6]SFWRTEX                                            22
F:        .FCTR [204,6]SFWRTIN                                            23
G:        .FCTR [204,6]SFWRTMG                                            24
H:        .FCTR [204,6]SFWRTMS                                            25
I:        .FCTR [204,6]SFWRTMT                                            26
J:        .FCTR [204,6]SFWRTPF                                            27
K:        .FCTR [204,6]SFWRTPP                                            28
L:        .FCTR [204,6]SFWRTPR                                            29
M:        .FCTR [204,6]SFWRTPS                                            30
N:        .FCTR [204,6]SFWRTRA                                            31
O:        .FCTR [204,6]SFWRTRK                                            32
P:        .FCTR [204,6]SFWRTSW                                            33
Q:        .FCTR [204,6]SFWRTTS                                            34
R:        .FCTR [204,6]SFWRTWF                                            35
S:        .FCTR [204,6]SFWRTYA                                            36
T:        .FCTR [204,6]SFWRTYE                                            37
U:        .FCTR [204,6]SFWRTYP                                            38
;                                                                         39
;                                                                         40
@LB:[1,1]RMS11M.ODL                                                       41
@LB:[1,1]RMS12X.ODL                                                       42
          .END                                                            43
```

Figure 3-42. DBRPTSEF Program Overlay Descriptor Language File (DBRPTSEF.ODL)

3.14 SUBJECTIVE EVALUATIONS DIRECTORY FILE LISTING PROCEDURE (DBRPTDIR)

3.14.1 INTRODUCTION

The Subjective Evaluations Directory File Listing Procedure (DBRPTDIR) lists the contents of the Subjective Evaluations Directory (DIR) file by using DATATRIEVE (Reference 4).

3.14.2 FILES ACCESSED

The DBRPTDIR procedure accesses one input file and one output file as described below.

Input File Name	Description
[204,1]DIR.HDR	Subjective Evaluations Directory File

Output File Name	Description
SEFDIR.RPT	Output listing file

3.14.3 DATATRIEVE COMMAND FILE

Figure 3-43 is a listing of DBRPTDIR.DTR under UIC [204,4], a DATATRIEVE command file that generates a listing of the contents of the DIR file.

```
SET DICTIONARY [204,1]QUERY.DIC;                                                1
READY SEFDIR;                                                                   2
FIND E IN SEFDIR SORTED BY CODE;                                                3
REPORT ALL CURRENT ON SEFDIR.RPT                                                4
SET REPORT-NAME="SUBJECTIVE EVALUATIONS DIRECTORY INFORMATION (DIR.HDR)"         5
    PRINT  CODE          ("CODE"),                                              6
           NAME          (" MEASURE"/"  NAME   "),                              7
           MIN-VALUE     ("  MIN "/" VALUE"),                                   8
           MAX-VALUE     ("  MAX "/" VALUE"),                                   9
           DATA-REC-NO   ("REC"/"SEQ"),                                        10
           BYTE-LOC      ("BYTE"/" LOC"),                                      11
           DESCRIPTION   ("DESCRIPTION")                                       12
REPORT END                                                                      13
                                                                                14
   YOUR REPORT IS ON FILES 'SEFDIR.RPT'                                         15
   PLEASE PRINT THIS FILE.                                                      16
                                                                                17
```

Figure 3-43. DBRPTDIR DATATRIEVE Command File (DBRPTDIR.DTR)

3.15 ENCODING DICTIONARY LISTING PROCEDURE (DBRPTENC)

3.15.1 INTRODUCTION

The Encoding Dictionary Listing Procedure (DBRPTENC) produces a listing of the contents of the Encoding Dictionary File by using DATATRIEVE (Reference 4).

3.15.2 FILES ACCESSED

The DBRPTENC procedure accesses one input file and one output file as described below.

Input File Name	Description
[204,1]ENCODE.HDR	Encoding Dictionary (ENC) file

Output File Name	Description
ENC.RPT	Output listing file of the Encoding Dictionary

3.15.3 DATATRIEVE COMMAND FILE

Figure 3-44 is a listing of DBRPTENC.DTR under UIC [204,4], the DATATRIEVE command file that generates a listing of the contents of the ENC File.

```
SET DICTIONARY [204,1]QUERY.DIC;                                    1
READY ENC;                                                          2
FIND E IN ENC;                                                      3
REPORT CURRENT SORTED BY TYPE, CODE ON ENC.RPT                      4
SET REPORT-NAME="ENCODING DICTIONARY (ENCODE.HDR)",LINES-PAGE=60,   5
    COLUMNS-PAGE=90                                                 6
AT TOP OF TYPE PRINT SKIP;                                          7
PRINT COL 10,TYPE USING ZZ9, COL 18,CODE USING XXXXX,               8
   COL 26, NAME, COL 40, REST                                       9
REPORT END;                                                        10
!                                                                  11
!  YOUR REPORT IS ON FILE 'ENC.RPT'                                12
```

Figure 3-44. DBRPTENC DATATRIEVE Command File (DBRPTENC.DTR)

3.16 PHASE DATES FILE LISTING PROCEDURE (DBRPTHDR)

3.16.1 INTRODUCTION

The Phase Dates File Listing Procedure (DBRPTHDR) produces a listing of the contents of the Phase Dates (HDR) file by using DATATRIEVE (Reference 4).

3.16.2 FILES ACCESSED

The DBRPTHDR procedure accesses one input file and one output file as described below.

Input File Name	Description
[204,1]HEADER.HDR	HDR file

Output File Name	Description
HDR.RPT	Output listing file of the HDR file

3.16.3 DATATRIEVE COMMAND FILE

Figure 3-45 is a listing of [204,4]DBRPTHDR.DTR, the DATATRIEVE command file that generates a listing of the contents of the HDR file.

```
SET DICTIONARY [204,1]QUERY.DIC;                                                1
READY HDR;                                                                      2
FIND S IN HDR WITH DES1 > 0                                                     3
REPORT ALL CURRENT SORTED BY NAME ON HDR.RPT                                    4
SET REPORT-NAME="HEADER DATA    (FILE [204,1]HEADER.HDR)",LINES-PAGE=65,        5
    COLUMNS-PAGE=78                                                             6
  PRINT NAME ("PROJECT"),                                                       7
        DES1    ("DESIGN"              /"START"/"DATE") USING ZZZZZZ,           8
        CODE1   ("CODE &"/"UNIT TEST" /"START"/"DATE") USING ZZZZZZ,           9
        SYS1    ("SYSTEM"/"TEST"       /"START"/"DATE") USING ZZZZZZ,          10
        ACC1    ("ACCEPTANCE"/"TEST"   /"START"/"DATE") USING ZZZZZZ,          11
        CLEAN1  ("CLEANUP"             /"START"/"DATE") USING ZZZZZZ,          12
        CLEAN2  ("CLEANUP"             /"END"  /"DATE") USING ZZZZZZ           13
REPORT END                                                                      14
|                                                                               15
|   YOUR REPORT IS ON FILE  'HDR.RPT'                                           16
```

Figure 3-45. DBRPTHDR DATATRIEVE Command File (DBRPTHDR.DTR)

3.17 FILE NAME AND STATUS FILE LISTING PROCEDURE (DBRPTSTS)

3.17.1 INTRODUCTION

The File Name and Status File Listing Procedure (DBRPTSTS) produces a listing of the contents of the File Name and Status (STS) file by using DATATRIEVE (Reference 4).

3.17.2 FILES ACCESSED

The DBRPTSTS procedure accesses one input file and one output file as described below.

Input File Name	Description
[204,1]STAT.HDR	STS file

Output File Name	Description
STAT.RPT	Output listing file of the STS file

3.17.3 DATATRIEVE COMMAND FILE

Figure 3-46 is a listing of [204,4]DBRPTSTS.DTR, the DATATRIEVE command file that generates a listing of the contents of the STS file.

```
SET DICTIONARY [204,1]QUERY.DIC                                          1
READY STAT                                                               2
FIND S IN STAT                                                           3
REPORT CURRENT SORTED BY PROJ ON STAT.RPT                                4
   SET REPORT-NAME="DIRECTORY FILE - STAT.DAT",                          5
      COLUMNS-PAGE=90                                                    6
   PRINT FILE USING ZZ, NAME, CREATE, BACKUP, UPDATE,                    7
      NREC USING ZZZZZ                                                   8
   AT TOP OF PROJ PRINT SKIP,"PROJECT =",PROJ USING ZZ                   9
REPORT END                                                              10
!                                                                       11
!  YOUR REPORT IS ON FILE STAT.RPT                                      12
```

Figure 3-46. DBRPTSTS DATATRIEVE Command File (DBRPTSTS.DTR)

3.18 ESTIMATED STATISTICS FILE LISTING PROCEDURE (DBRPTEST)

3.18.1 INTRODUCTION

The Estimated Statistics File Listing Procedure (DBRPTEST) produces a listing of the contents of the Estimated Statistics (EST) file by using DATATRIEVE (Reference 4).

3.18.2 FILES ACCESSED

The DBRPTEST procedure accesses one input file and two output files as described below.

Input File Name	Description
[204,1]EST.HDR	EST file

Output File Name	Description
EST1.RPT	Part one of the output report of the contents of the EST file
EST2.RPT	Part two of the output report of the contents of the EST file

3.18.3 DATATRIEVE COMMAND FILE

Figure 3-47 is a listing of DBRPTEST.DTR under UIC [204,4], the DATATRIEVE command file that generates the listings of the contents of the EST file.

```
SET DICTIONARY [204,1]QUERY.DIC;                                          1
READY ESTX;                                                               2
FIND E IN ESTX SORTED BY NAME;                                            3
REPORT ALL CURRENT ON EST1.RPT                                            4
SET REPORT-NAME="ESTIMATED STATISTICS -- PART 1"                          5
   PRINT  NAME           ("PROJECT"),                                     6
          PROJ           ("PROJ"/"CODE"),                                 7
          COMPN          (" #" / "COMP"),                                 8
          TOT-MOD        (" TOT"/" #"/" MOD"),                            9
          NEW-MOD        (" #"/" NEW"/" MOD"),                           10
          MOD-MOD        (" #"/" MOD"/" MOD"),                           11
          RUNS           ("  # OF" / "  RUNS"),                          12
          CHANGES        (" # OF" / "CHANGES"),                          13
          DOC            (" PAGES"/" OF"/" DOC"),                        14
          TOTAL-LINES    (" TOTAL"/" # OF"/" LINES"),                    15
          NEW-LINES      (" # OF"/" NEW"/" LINES"),                      16
          MOD-LINES      (" # OF"/" MODIF"/" LINES"),                    17
          TOTAL-EXEC     ("# OF"/" TOTAL"/"EXECUT"),                     18
          NEW-EXEC       ("# OF"/" NEW"/"EXECUT"),                       19
          MOD-EXEC       ("# OF"/" MODIF"/"EXECUT")                      20
REPORT END                                                                21
|                                                                         22
REPORT ALL CURRENT ON EST2.RPT                                            23
SET REPORT-NAME="ESTIMATED STATISTICS -- PART 2"                          24
   PRINT  NAME           ("PROJECT"),                                     25
          PROJ           ("PROJ"/"CODE"),                                 26
          PROG           ("  PROG" / " HOURS"),                          27
          MGMT           ("  MGMT" / " HOURS"),                          28
          OTHER          (" OTHER" / " HOURS"),                          29
          S95            ("  360 " / "   95 " / " HOURS"),               30
          S75            ("  360 " / "   75 " / " HOURS"),               31
          OTH            (" OTHER"/"CMPUTR"/" HOURS"),                   32
          STATUS         ("STAT"/"FLAG"),                                33
          ACTIVE         ("ACTV"/"FL AG"),                               34
          PROJ-CATEGORY ("PROJ"/" CATG")                                 35
REPORT END                                                                36
|                                                                         37
|  YOUR REPORT IS ON FILES 'EST1.RPT' FOR PART 1 AND                      38
|                          'EST2.RPT' FOR PART 2.                         39
|  PLEASE PRINT THESE FILES.                                              40
|                                                                         41
```

Figure 3-47. DBRPTEST DATATRIEVE Command File (DBRPTEST.DTR)

REFERENCES

1. Software Engineering Laboratory, Software Engineering Laboratory (SEL) Data BAse Organization and User's Guide, SEL-81-002, D. C. Wyckoff, G. Page, F. E. McGarry, et al., September 1981

2. --, Software Engineering Laboratory (SEL) Data Base Maintenance System (DBAM) User's Guide and System Description, SEL-81-003, D. N. Card, D. C. Wyckoff, G. Page, et al., September 1981

3. --, Evaluation of Management Measures of Software Development, SEL-82-001, Volumes 1 and 2, D. N. Card, G. Page, and F. E. McGarry, September 1982

4. Digital Equipment Corporation, User's Guide to DATATRIEVE-11, December 1977

BIBLIOGRAPHY OF SEL LITERATURE

The technical papers, memorandums, and documents listed in this bibliography are organized into two groups. The first group is composed of documents issued by the Software Engineering Laboratory (SEL) during its research and development activities. The second group includes materials that were published elsewhere but pertain to SEL activities.

SEL-Originated Documents

SEL-76-001, Proceedings From the First Summer Software Engineering Workshop, August 1976

SEL-77-001, The Software Engineering Laboratory, V. R. Basili, M. V. Zelkowitz, F. E. McGarry, et al., May 1977

SEL-77-002, Proceedings From the Second Summer Software Engineering Workshop, September 1977

SEL-77-003, Structured FORTRAN Preprocessor (SFORT), B. Chu and D. S. Wilson, September 1977

SEL-77-004, GSFC NAVPAK Design Specifications Languages Study, P. A. Scheffer and C. E. Velez, October 1977

SEL-78-001, FORTRAN Static Source Code Analyzer (SAP) Design and Module Descriptions, E. M. O'Neill, S. R. Waligora, and C. E. Goorevich, February 1978

†SEL-78-002, FORTRAN Static Source Code Analyzer (SAP) User's Guide, E. M. O'Neill, S. R. Waligora, and C. E. Goorevich, February 1978

SEL-78-102, FORTRAN Static Source Code Analyzer Program (SAP) User's Guide (Revision 1), W. J. Decker and W. A. Taylor, September 1982

SEL-78-003, Evaluation of Draper NAVPAK Software Design, K. Tasaki and F. E. McGarry, June 1978

†This document superseded by revised document.

SEL-78-004, Structured FORTRAN Preprocessor (SFORT) PDP-11/70 User's Guide, D. S. Wilson and B. Chu, September 1978

SEL-78-005, Proceedings From the Third Summer Software Engineering Workshop, September 1978

SEL-78-006, GSFC Software Engineering Research Requirements Analysis Study, P. A. Scheffer and C. E. Velez, November 1978

SEL-78-007, Applicability of the Rayleigh Curve to the SEL Environment, T. E. Mapp, December 1978

SEL-79-001, SIMPL-D Data Base Reference Manual, M. V. Zelkowitz, July 1979

SEL-79-002, The Software Engineering Laboratory: Relationship Equations, K. Freburger and V. R. Basili, May 1979

SEL-79-003, Common Software Module Repository (CSMR) System Description and User's Guide, C. E. Goorevich, A. L. Green, and S. R. Waligora, August 1979

SEL-79-004, Evaluation of the Caine, Farber, and Gordon Program Design Language (PDL) in the Goddard Space Flight Center (GSFC) Code 580 Software Design Environment, C. E. Goorevich, A. L. Green, and W. J. Decker, September 1979

SEL-79-005, Proceedings From the Fourth Summer Software Engineering Workshop, November 1979

SEL-80-001, Functional Requirements/Specifications for Code 580 Configuration Analysis Tool (CAT), F. K. Banks, A. L. Green, and C. E. Goorevich, February 1980

SEL-80-002, Multi-Level Expression Design Language-Requirement Level (MEDL-R) System Evaluation, W. J. Decker and C. E. Goorevich, May 1980

SEL-80-003, Multimission Modular Spacecraft Ground Support Software System (MMS/GSSS) State-of-the-Art Computer Systems/Compatibility Study, T. Welden, M. McClellan, and P. Liebertz, May 1980

†SEL-80-004, System Description and User's Guide for Code 580 Configuration Analysis Tool (CAT), F. K. Banks, W. J. Decker, J. G. Garrahan, et al., October 1980

---

†This document superseded by revised document.

SEL-80-104, Configuration Analysis Tool (CAT) System Description and User's Guide (Revision 1), W. Decker and W. Taylor, December 1982

SEL-80-005, A Study of the Musa Reliability Model, A. M. Miller, November 1980

SEL-80-006, Proceedings From the Fifth Annual Software Engineering Workshop, November 1980

SEL-80-007, An Appraisal of Selected Cost/Resource Estimation Models for Software Systems, J. F. Cook and F. E. McGarry, December 1980

†SEL-81-001, Guide to Data Collection, V. E. Church, D. N. Card, F. E. McGarry, et al., September 1981

SEL-81-101, Guide to Data Collection, V. E. Church, D. N. Card, F. E. McGarry, et al., August 1982

†SEL-81-002, Software Engineering Laboratory (SEL) Data Base Organization and User's Guide, D. C. Wyckoff, G. Page, and F. E. McGarry, September 1981

†SEL-81-102, Software Engineering Laboratory (SEL) Data Base Organization and User's Guide Revision 1, P. Lo and D. Wyckoff, March 1983 (superseded by July 1983 version of SEL-81-102)

†SEL-81-003, Data Base Maintenance System (DBAM) User's Guide and System Description, D. N. Card, D. C. Wyckoff, and G. Page, September 1981

†SEL-81-103, Software Engineering Laboratory (SEL) Data Base Maintenance System (DBAM) User's Guide and System Description, P. Lo and D. Card, April 1983 (superseded by July 1983 version of SEL-81-103)

†SEL-81-004, The Software Engineering Laboratory, D. N. Card, F. E. McGarry, G. Page, et al., September 1981

SEL-81-104, The Software Engineering Laboratory, D. N. Card, F. E. McGarry, G. Page, et al., February 1982

†SEL-81-005, Standard Approach to Software Development, V. E. Church, F. E. McGarry, G. Page, et al., September 1981

SEL-81-105, Recommended Approach to Software Development, S. Eslinger, F. E. McGarry, and G. Page, May 1982

†This document superseded by revised document.

SEL-81-006, Software Engineering Laboratory (SEL) Document Library (DOCLIB) System Description and User's Guide, W. Taylor and W. J. Decker, December 1981

†SEL-81-007, Software Engineering Laboratory (SEL) Compendium of Tools, W. J. Decker, E. J. Smith, A. L. Green, et al., February 1981

SEL-81-107, Software Engineering Laboratory (SEL) Compendium of Tools, W. J. Decker, W. A. Taylor, and E. J. Smith, February 1982

SEL-81-008, Cost and Reliability Estimation Models (CAREM) User's Guide, J. F. Cook and E. Edwards, February 1981

SEL-81-009, Software Engineering Laboratory Programmer Workbench Phase 1 Evaluation, W. J. Decker and F. E. McGarry, March 1981

SEL-81-010, Performance and Evaluation of an Independent Software Verification and Integration Process, G. Page and F. E. McGarry, May 1981

SEL-81-011, Evaluating Software Development by Analysis of Change Data, D. M. Weiss, November 1981

SEL-81-012, The Rayleigh Curve As a Model for Effort Distribution Over the Life of Medium Scale Software Systems, G. O. Picasso, December 1981

SEL-81-013, Proceedings From the Sixth Annual Software Engineering Workshop, December 1981

SEL-81-014, Automated Collection of Software Engineering Data in the Software Engineering Laboratory (SEL), A. L. Green, W. J. Decker, and F. E. McGarry, September 1981

SEL-82-001, Evaluation of Management Measures of Software Development, G. Page, D. N. Card, and F. E. McGarry, September 1982, vols. 1 and 2

SEL-82-002, FORTRAN Static Source Code Analyzer Program (SAP) System Description, W. A. Taylor and W. J. Decker, August 1982

SEL-82-003, Software Engineering Laboratory (SEL) Data Base Reporting Software User's Guide and System Description, P. Lo, September 1982 (superseded by August 1983 version of SEL-82-003)

---

†This document superseded by revised document.

SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982

SEL-82-005, Glossary of Software Engineering Laboratory Terms, M. G. Rohleder, December 1982

SEL-82-006, Annotated Bibliography of Software Engineering Laboratory (SEL) Literature, D. N. Card, November 1982

SEL-82-007, Proceedings From the Seventh Annual Software Engineering Workshop, December 1982

SEL-82-008, Evaluating Software Development by Analysis of Changes: The Data From the Software Engineering Laboratory, V. R. Basili and D. M. Weiss, December 1982

SEL-Related Literature

††Bailey, J. W., and V. R. Basili, "A Meta-Model for Software Development Resource Expenditures," Proceedings of the Fifth International Conference on Software Engineering. New York: Computer Societies Press, 1981

Banks, F. K., "Configuration Analysis Tool (CAT) Design," Computer Sciences Corporation, Technical Memorandum, March 1980

††Basili, V. R., "Models and Metrics for Software Management and Engineering," ASME Advances in Computer Technology, January 1980, vol. 1

Basili, V. R., "SEL Relationships for Programming Measurement and Estimation," University of Maryland, Technical Memorandum, October 1979

Basili, V. R., Tutorial on Models and Metrics for Software Management and Engineering. New York: Computer Societies Press, 1980 (also designated SEL-80-008)

††Basili, V. R., and J. Beane, "Can the Parr Curve Help With Manpower Distribution and Resource Estimation Problems?", Journal of Systems and Software, February 1981, vol. 2, no. 1

---

††This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982.

††Basili, V. R., and K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory," Journal of Systems and Software, February 1981, vol. 2, no. 1

Basili, V. R., and B. T. Perricone, Software Errors and Complexity: An Empirical Investigation, University of Maryland, Technical Report TR-1195, August 1982

††Basili, V. R., and T. Phillips, "Evaluating and Comparing Software Metrics in the Software Engineering Laboratory," Proceedings of the ACM SIGMETRICS Symposium/Workshop: Quality Metrics, March 1981

Basili, V. R., R. W. Selby, and T. Phillips, Metric Analysis and Data Validation Across FORTRAN Projects, University of Maryland, Technical Report, November 1982

Basili, V. R., and R. Reiter, "Evaluating Automatable Measures for Software Development," Proceedings of the Workshop on Quantitative Software Models for Reliability, Complexity and Cost, October 1979

Basili, V.R., and D. M. Weiss, A Methodology for Collecting Valid Software Engineering Data, University of Maryland, Technical Report TR-1235, December 1982

Basili, V. R., and M. V. Zelkowitz, "Designing a Software Measurement Experiment," Proceedings of the Software Life Cycle Management Workshop, September 1977

††Basili, V. R., and M. V. Zelkowitz, "Operation of the Software Engineering Laboratory," Proceedings of the Second Software Life Cycle Management Workshop, August 1978

††Basili, V. R., and M. V. Zelkowitz, "Measuring Software Development Characteristics in the Local Environment," Computers and Structures, August 1978, vol. 10

Basili, V. R., and M. V. Zelkowitz, "Analyzing Medium Scale Software Development," Proceedings of the Third International Conference on Software Engineering. New York: Computer Societies Press, 1978

---

††This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982.

††Basili, V. R., and M. V. Zelkowitz, "The Software Engineering Laboratory: Objectives," Proceedings of the Fifteenth Annual Conference on Computer Personnel Research, August 1977

Card, D. N., "Early Estimation of Resource Expenditures and Program Size," Computer Sciences Corporation, Technical Memorandum, June 1982

Card, D. N., "Comparison of Regression Modeling Techniques for Resource Estimation," Computer Sciences Corporation, Technical Memorandum, November 1982

Card, D. N., and M. G. Rohleder, "Report of Data Expansion Efforts," Computer Sciences Corporation, Technical Memorandum, September 1982

††Chen, E., and M. V. Zelkowitz, "Use of Cluster Analysis To Evaluate Software Engineering Methodologies," Proceedings of the Fifth International Conference on Software Engineering. New York: Computer Societies Press, 1981

Freburger, K., "A Model of the Software Life Cycle" (paper prepared for the University of Maryland, December 1978)

Higher Order Software, Inc., TR-9, A Demonstration of AXES for NAVPAK, M. Hamilton and S. Zeldin, September 1977 (also designated SEL-77-005)

Hislop, G., "Some Tests of Halstead Measures" (paper prepared for the University of Maryland, December 1978)

Lange, S. F., "A Child's Garden of Complexity Measures" (paper prepared for the University of Maryland, December 1978)

Miller, A. M., "A Survey of Several Reliability Models" (paper prepared for the University of Maryland, December 1978)

National Aeronautics and Space Administration (NASA), NASA Software Research Technology Workshop (proceedings), March 1980

Page, G., "Software Engineering Course Evaluation," Computer Sciences Corporation, Technical Memorandum, December 1977

---

††This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982.

Parr, F., and D. Weiss, "Concepts Used in the Change Report Form," NASA, Goddard Space Flight Center, Technical Memorandum, May 1978

Reiter, R. W., "The Nature, Organization, Measurement, and Management of Software Complexity" (paper prepared for the University of Maryland, December 1976)

Scheffer, P. A., and C. E. Velez, "GSFC NAVPAK Design Higher Order Languages Study: Addendum," Martin Marietta Corporation, Technical Memorandum, September 1977

Turner, C., and G. Caron, A Comparison of RADC and NASA/SEL Software Development Data, Data and Analysis Center for Software, Special Publication, May 1981

Turner, C., G. Caron, and G. Brement, NASA/SEL Data Compendium, Data and Analysis Center for Software, Special Publication, April 1981

Weiss, D. M., "Error and Change Analysis," Naval Research Laboratory, Technical Memorandum, December 1977

Williamson, I. M., "Resource Model Testing and Information," Naval Research Laboratory, Technical Memorandum, July 1979

††Zelkowitz, M. V., "Resource Estimation for Medium Scale Software Projects," Proceedings of the Twelfth Conference on the Interface of Statistics and Computer Science. New York: Computer Societies Press, 1979

Zelkowitz, M. V., "Data Collection and Evaluation for Experimental Computer Science Research," Empirical Foundations for Computer and Information Science (proceedings), November 1982

Zelkowitz, M. V., and V. R. Basili, "Operational Aspects of a Software Measurement Facility," Proceedings of the Software Life Cycle Management Workshop, September 1977

††This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982.